

---

**RPClass - Aplicación para gamificación del aula**  
**RPClass - Application for class gamification**

---



**Trabajo de Fin de Máster**  
**Curso 2019–2020**

**Autor**  
**Aitor Cayón Ruano**

**Director**  
**Manuel Freire Morán**

**Máster en Ingeniería Informática**  
**Facultad de Informática**  
**Universidad Complutense de Madrid**



# **RPClass - Aplicación para gamificación del aula RPClass - Application for class gamification**

**Trabajo de Fin de Máster en Ingeniería Informática  
Departamento de Ingeniería del Software e Inteligencia Artificial**

**Autor  
Aitor Cayón Ruano**

**Director  
Manuel Freire Morán**

**Convocatoria: Junio-Julio 2020  
Calificación: 6.5**

**Máster en Ingeniería Informática  
Facultad de Informática  
Universidad Complutense de Madrid**

**2 de julio de 2020**



# Dedicatoria

*A Noé y Rosa, por darme fuerzas para llegar  
donde estoy*



# Agradecimientos

A Manuel, por su tiempo, dedicación y compromiso durante la dirección del proyecto. Y a mi familia, por su constante apoyo a pesar de la distancia.





# Resumen

## **RPClass - Aplicación para gamificación del aula**

Hoy en día la tecnología está cada vez más presente en la educación, y la enseñanza se encuentra en constante evolución para adaptarse e integrar los últimos avances tecnológicos. Entre los progresos en e-learning que más relevancia y popularidad han adquirido en los últimos años destaca la gamificación, entendida como la aplicación de mecánicas de juegos al ámbito educativo-profesional. El uso de gamificación ha demostrado efectos positivos sobre la motivación, y por lo tanto una mayor atención y mejor aprendizaje.

**RPClass** es una aplicación web diseñada para la gamificación de actividades en el aula. Permite organizar competiciones en las que los estudiantes, organizados por equipos, deberán superar las pruebas que se les planteen para alcanzar el puesto más alto del ranking. Esto a su vez permitirá a los alumnos desbloquear varios logros, tanto individuales como grupales, como recompensa a su esfuerzo.

Esta plataforma permite a los profesores que la usen generar perfiles para sus alumnos, organizar equipos y concursos, y crear pruebas asociadas, que serán evaluadas de forma automática, generándose resultados listos para su análisis. El proyecto ha sido diseñado como una herramienta de apoyo a la labor docente, ayudando a la evaluación por parte del profesor de los conocimientos adquiridos, y como un medio para promover la motivación e interacción entre los alumnos. A través de esta aplicación se espera conseguir clases más dinámicas y participativas fomentando los valores de constancia, competitividad y trabajo en equipo.

## **Palabras clave**

Gamificación, aplicación web, motivación, aprendizaje, e-learning, blended learning, competición, logros, rankings



# Abstract

## **RPClass - Application for class gamification**

Technology is becoming ever more present in education, and teaching is constantly evolving to adapt and integrate the latest technological innovations. Among popular e-learning innovations, gamification stands out, understood as the application of game mechanics to educational-professional fields. The use of gamification has shown positive effects on motivation, promoting greater attention and better learning.

**RPClass** is a web application designed for classroom gamification. It allows organizing competitions in which students, grouped in teams, will have to pass contests to reach the highest position in the ranking. At the same time this will allow students to unlock several individual and group achievements as a reward to their effort.

This platform allows teachers using it to generate profiles for their students, organize teams and contests and create tests, which will be automatically graded, generating results ready to be analyzed. The project has been designed as a supporting tool for educational work, helping teachers to evaluate the acquired knowledge, and as a means to improve motivation and interaction among students. Through this application, we expect to achieve more dynamic and participatory classes while promoting values of perseverance, competitiveness and teamwork.

## **Keywords**

Gamification, web application, motivation, learning, e-learning, blended learning, competition, achievements, rankings



# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	3
1.3. Modelo de desarrollo . . . . .	3
1.4. Plan de trabajo . . . . .	4
<b>2. Gamificación y tecnología en la educación</b>	<b>7</b>
2.1. Gamificación . . . . .	7
2.2. Mecánicas de gamificación . . . . .	8
2.3. Experimentos y proyectos . . . . .	10
2.4. Tecnologías . . . . .	11
<b>3. Descripción del Trabajo</b>	<b>13</b>
3.1. Especificación de requisitos . . . . .	13
3.2. Diseño . . . . .	14
3.2.1. Análisis de tecnologías: vista . . . . .	15
3.2.2. Análisis de tecnologías: modelo y controlador . . . . .	17
3.2.3. Modelo de datos . . . . .	20
3.2.4. Interfaz . . . . .	22
3.2.5. Inicio de sesión . . . . .	24
3.2.6. Carga de datos . . . . .	24

3.3. Implementación . . . . .	26
3.3.1. Inicio de sesión . . . . .	27
3.3.2. Creación de clases . . . . .	29
3.3.3. Creación de pruebas . . . . .	30
3.3.4. Participación en pruebas . . . . .	31
3.3.5. Generación de rankings . . . . .	33
3.3.6. Logros y progresión . . . . .	34
3.4. Pruebas y testeo de la aplicación . . . . .	35
3.4.1. Caso de uso: creación de una clase . . . . .	35
3.4.2. Caso de uso: creación de una prueba . . . . .	37
3.4.3. Caso de uso: participación en una prueba . . . . .	38
<b>4. Conclusiones y Trabajo Futuro</b>	<b>39</b>
<b>5. Introduction</b>	<b>45</b>
5.1. Motivation . . . . .	45
5.2. Objectives . . . . .	46
5.3. Development model . . . . .	47
5.4. Work plan . . . . .	48
<b>6. Conclusions and Future Work</b>	<b>51</b>
<b>A. Manual de instalación</b>	<b>57</b>
A.1. Configuración del directorio de ficheros . . . . .	57
A.2. Configuración de la base de datos . . . . .	57
A.3. Ejecución de la aplicación desde Eclipse . . . . .	58
<b>B. Manual de uso</b>	<b>61</b>
B.1. Creación de una clase . . . . .	61
B.2. Creación de una prueba . . . . .	63
B.3. Gestión de las pruebas . . . . .	66







# Índice de figuras

1.1. Flujo de un modelo de desarrollo iterativo e incremental. Wikimedia Commons (2014) . . . . .	4
2.1. Pirámide de las necesidades de Maslow, 1943. Wikimedia Commons (2006)	8
2.2. Pirámide de las necesidades del jugador. Ang y Rao (2004) . . . . .	9
2.3. Ejemplo de logro en el portal web de Just Press Play. Elizabeth Lawley (2012) . . . . .	10
2.4. Comparación de LMS. Cujba (2020) . . . . .	12
3.1. Estadísticas de popularidad de Angular, React y Vue. (Daityari, 2019) . .	15
3.2. Estadísticas de uso de frameworks de CSS. (Watson-Nolan, 2019) . . . .	16
3.3. Estadísticas de uso de PHP, .NET y Java en aplicaciones web. (W3Techs, 2020) . . . . .	18
3.4. Estadística de uso de frameworks en Java. (JetBrains, 2019) . . . . .	19
3.5. Paquetes incluidos en Spring Framework. Spring (2016) . . . . .	20
3.6. Modelo de datos . . . . .	21
3.7. Ejemplo de boceto inicial con navegación mediante menú lateral . . . . .	22
3.8. Ejemplo de funcionalidad común. Resolución de pruebas . . . . .	23
3.9. Controladores, modelo de datos y herramientas del sistema . . . . .	26
3.10 Flujo completo de ejecución de la plataforma . . . . .	27
3.11 Detalle de un fichero con las credenciales de acceso . . . . .	28
3.12 Creación de los equipos a través de la vista . . . . .	29
3.13 Resultados y estadísticas de una prueba . . . . .	31

3.14	Rankings global de estudiantes y equipos . . . . .	33
3.15	Lista de logros de un alumno y su progreso . . . . .	36
4.1.	Lenguajes utilizados en el proyecto y sus líneas de código . . . . .	39
5.1.	Flow of an iterative and incremental development model. Wikimedia Commons (2014) . . . . .	48
6.1.	programming languages used in the project and lines of code . . . . .	51
A.1.	Localización de IwApplication.java en el proyecto . . . . .	59
A.2.	Acceso a la plataforma desde el navegador . . . . .	59
B.1.	Fichero de clase seleccionado . . . . .	62
B.2.	Vista de clases tras la carga de fichero . . . . .	62
B.3.	Organización de los alumnos en equipos . . . . .	63
B.4.	Vista de clases tras completar el proceso de creación . . . . .	63
B.5.	Fichero de prueba seleccionado . . . . .	65
B.6.	Prueba cargada y añadida a la correspondiente clase . . . . .	65
B.7.	Vista de una prueba recién creada . . . . .	66
B.8.	Resultados de una prueba y estadísticas de las preguntas . . . . .	66
B.9.	Rankings de estudiantes y equipos en una prueba . . . . .	67

# Introducción

La tecnología siempre se ha utilizado como herramienta de apoyo para la educación, aportando nuevos enfoques y facilitando su acceso a aquellos con mayores dificultades (Raja y Nagasubramani, 2018). Es una realidad presente en la mayoría de nuestras aulas y su implantación es cada vez más extensa. Actualmente es posible incluso formarse desde casa gracias a la gran cantidad de recursos digitales disponibles, pudiendo acceder fácilmente al conocimiento desde una gran variedad de plataformas.

Aprovechando el potencial de la tecnología han surgido múltiples alternativas para apoyar el aprendizaje, desde cursos online hasta grandes plataformas de e-learning. Desde mi experiencia cómo alumno he podido comprobar que el ambiente del aula cambia cuando se realiza alguna actividad que rompa con la rutina. Con frecuencia estas están relacionadas con la utilización de algún dispositivo electrónico o software. La interacción entre los compañeros de clase es mayor y su participación es más frecuente de lo habitual, especialmente si la actividad en curso implica un factor competitivo. Durante los estudios del grado algunas de estas experiencias han consistido en encuestas con *clickers*, concursos de Kahoot! y competiciones clasificatorias para la resolución de las prácticas de una asignatura. La propuesta de este proyecto se enmarca en el ámbito de la gamificación, la aplicación de mecánicas de juegos al ámbito educativo-profesional con el fin de mejorar la motivación y el rendimiento.

## 1.1. Motivación

De entre las múltiples aplicaciones de la tecnología en la educación una de las que más interés genera es la gamificación. Destaca principalmente por proponer un enfoque de la enseñanza más estimulante, en el que se recurre a mecánicas de juegos con el fin de mejorar los resultados obtenidos. Adaptando las características de los juegos a la educación se consigue una aproximación más práctica, interactiva y divertida (Gee, 2005). A partir de la gamificación es posible implementar nuevas formas de trabajar, explorar y hacer más atractivo el material docente.

Un factor fundamental a tener en cuenta a la hora de enseñar es mantener la motivación de los estudiantes. Las nuevas tecnologías han demostrado su utilidad a la hora de convertir el aprendizaje en una tarea más atractiva y accesible, y por tanto más motivadora (Francis, 2017). Cada vez es más frecuente la utilización de dispositivos electrónicos como complemento a las metodologías tradicionales de enseñanza. Numerosos estudios han analizado si la utilización de tecnologías durante la formación conlleva mejores resultados y un incremento del interés y la implicación por parte de los alumnos (Hainey et al., 2016).

La combinación de la enseñanza presencial, en la que los alumnos reciben lecciones de un profesor en una clase, con la utilización de dispositivos electrónicos y recursos educativos online se conoce como *blended learning* o *b-learning* (Bartolomé, 2004). Aprovechando las ventajas de ambas modalidades es posible extender la educación más allá del aula y facilitar el acceso de los alumnos al material docente. Sin embargo, también es necesario tener en cuenta la inversión temporal y de recursos para formar tecnológicamente al profesorado en la organización y gestión de un modelo de enseñanza con *blended learning*.

**RPClass** toma este escenario como base y propone la gamificación de las clases a través de una aplicación web. Desde esta plataforma el profesor podrá preparar pruebas que sus alumnos podrán resolver desde cualquier dispositivo con acceso a Internet, a la vez que compiten por conseguir los mejores resultados. Partiendo de esta premisa, el proyecto se ha implementado como una herramienta experimental de apoyo a la labor del profesorado y medio para estimular la participación de los estudiantes.

La aplicación puede ser de especial utilidad en materias que requieran de la memorización de una gran cantidad de información teórica. Al no precisar de la resolución de problemas, adaptar las pruebas al sistema es una tarea más asequible. La gamificación de estas materias puramente teóricas puede aportar un enfoque más dinámico y participativo, evitando la monotonía y pérdida de motivación de los estudiantes. Mediante la organización de pruebas puede animarse al alumno a adquirir hábitos de estudio continuo y comprobar si la clase está asimilando correctamente el contenido impartido en el aula.

Una de las características presentes en **RPClass**, a través de la cual se espera promover el apoyo mutuo entre los alumnos durante el proceso de aprendizaje, es el enfoque cooperativo dentro de la gamificación propuesta. Los estudiantes, organizados en grupos, deberán colaborar para que el equipo alcance el mejor puesto en la clasificación. Mediante el planteamiento de una meta común se espera un esfuerzo colectivo en el que cada alumno, junto a sus compañeros, intente progresar y obtener buenos resultados. El resultado final dependerá de la implicación de cada integrante. Estudios han demostrado que al permitir que los alumnos se gestionen entre si motiva la justa evaluación de las aportaciones individuales en el grupo (Moccozet et al., 2013).

## 1.2. Objetivos

La finalidad de este proyecto es el desarrollo de una aplicación web para gamificación de ejercicios en el aula. Para ello se tomarán las distintas pruebas que podrían utilizar los docentes como método de evaluación y se les dará un nuevo enfoque. La resolución de estas se trasladará a una plataforma web y los estudiantes que las completen recibirán una puntuación en función de su desempeño. Los alumnos se organizarán en equipos y participarán en una competición para conseguir puntos y desbloquear logros. Cada logro reconoce el cumplimiento de unas metas determinadas, por ejemplo, el alcanzar una cantidad de pruebas superadas o preguntas respondidas correctamente. Para cada prueba se recopilarán los resultados de los alumnos y se generarán las gráficas con las estadísticas de las respuesta de cada pregunta.

El sistema está pensado para su utilización en aulas universitarias y de instituto con la tecnología ya disponible para docentes y estudiantes. El escenario más habitual se espera que sea el del profesor preparando una actividad desde su PC, y los alumnos resolviéndola desde sus respectivos ordenadores o dispositivos portátiles. También se contempla el escenario en el que el docente cree una prueba y la deja disponible para que sus alumnos puedan resolverla en casa.

Teniendo en cuenta el anterior planteamiento de la aplicación y los escenarios más frecuentes, la plataforma deberá cumplir los siguientes objetivos:

- Fácil interacción con el sistema: el profesorado debe poder crear de forma sencilla sus propias clases y pruebas, así como formar los equipos en los que se organizarán los estudiantes.
- Resolución de pruebas: los alumnos deben poder participar desde la plataforma en las pruebas creadas por los docentes. Controlar la disponibilidad de las mismas.
- Evaluación de los resultados: para simplificar la labor de los docentes las pruebas deben corregirse automáticamente. Además debe facilitarse el seguimiento del progreso de la clase.

## 1.3. Modelo de desarrollo

El modelo de desarrollo que mejor se adapta a la naturaleza y características del proyecto es el iterativo e incremental (Fig. 1.1). Este nos permitirá una adaptación sencilla a los potenciales cambios que vayan surgiendo así como la obtención de resultados a corto plazo con cada iteración. Dentro del proyecto puede realizarse una clara división de acuerdo a los dos roles de usuario fundamentales, la lógica de los profesores y la de los estudiantes. Adicionalmente, podemos clasificar las implementaciones de acuerdo a los elementos del modelo de datos que intervengan.

Dado este planteamiento, el desarrollo se estructurará en tres iteraciones. La primera abarcará la fase de diseño, la segunda se corresponderá con la implementación

y durante la tercera se elaborará la documentación del proyecto. La segunda iteración constará de varios ciclos al final de los cuales se obtendrá un prototipo con nuevas características funcionales y testeadas. A través de esta metodología de trabajo se pretende conseguir la mayor estabilidad posible en el proyecto. Acotando y completando funcionalidades antes de avanzar al siguiente ciclo se simplifica el control de versiones. De este modo, ante un error imprevisto será más sencillo la identificación y corrección del mismo consiguiendo un desarrollo más fluido y minimizando los riesgos de bloqueo.

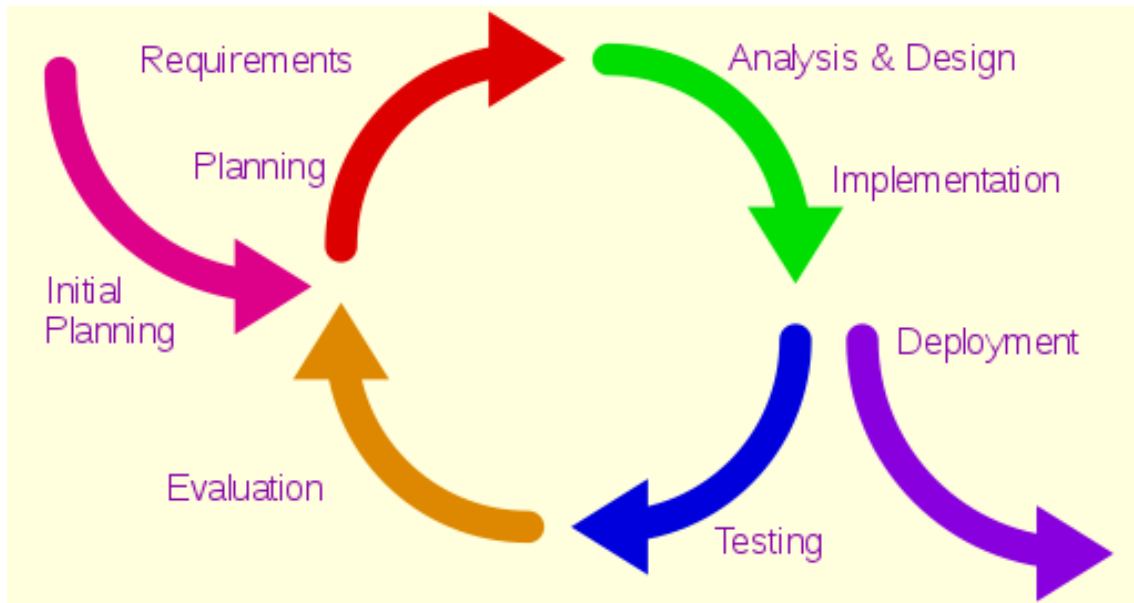


Figura 1.1: Flujo de un modelo de desarrollo iterativo e incremental. Wikimedia Commons (2014)

## 1.4. Plan de trabajo

Dada la falta de experiencia previa en este tipo de proyectos, y en previsión de posibles dificultades que puedan surgir durante el desarrollo, la mayor parte del tiempo disponible se destinará a la fase de implementación. Para minimizar los riesgos, la fase de diseño se extenderá hasta que se hayan definido adecuadamente todos los factores del desarrollo. Finalmente, el tiempo restante se destinará a la documentación del proyecto y la preparación para su distribución. Teniendo en cuenta este escenario, la planificación estimaría los siguientes plazos aproximados por fase y tareas a desempeñar:

- Fase de diseño:
  - Duración: 4 semanas. 04/02/20 a 02/03/20.
  - Ciclo 1: Especificación de requisitos: 1 semana
    - Tareas:
      - ◊ Definición de objetivos y requisitos

- ◊ Definición de la arquitectura y componentes del sistema
- Ciclo 2: Selección de tecnologías: 1 semana
  - Tareas:
    - ◊ Análisis y selección de tecnologías sistema
- Ciclo 3: Diseño del apartado gráfico: 1 semana
  - Tareas:
    - ◊ Diseño de las vistas
    - ◊ Diseño de la navegación e interacciones
    - ◊ Elaboración de *mockups*
- Ciclo 4: Diseño del modelo de datos: 1 semana
  - Tareas:
    - ◊ Definición de las clases del modelo de datos
- Fase de implementación:
  - Duración: 8 semanas. 03/03/20 a 27/04/20
  - Ciclo 1: Creación de clases: 2 semanas
    - Tareas:
      - ◊ Procesado de ficheros
      - ◊ Organización de equipos
  - Ciclo 2: Creación de pruebas: 2 semanas
    - Tareas:
      - ◊ Procesado de ficheros
      - ◊ Prueba disponible para resolver
  - Ciclo 3: Participación en pruebas: 2 semanas
    - Tareas:
      - ◊ Recolección de resultados
      - ◊ Generación de las gráficas
      - ◊ Generación del ranking
  - Ciclo 4: Incorporación de logros: 1 semana
    - Tareas:
      - ◊ Creación de logros
      - ◊ Actualización tras las pruebas
  - Ciclo 5: Pruebas del sistema: 1 semana
    - Tareas:
      - ◊ Revisión de casos de uso
      - ◊ Ajustes de interfaz
      - ◊ Experimento con usuarios reales
- Fase de documentación:
  - Duración: 6 semanas. 28/04/20 a 08/06/20

- Ciclo 1: Elaboración de la memoria: 3 semanas
  - Tareas:
    - ◊ Redacción de la memoria
    - ◊ Recopilación de la bibliografía
    - ◊ Generación de diagramas
- Ciclo 2: Documentación del código: 1 semana
  - Tareas:
    - ◊ Comentarios en el código
    - ◊ Actualización de la versión definitiva en el repositorio
- Ciclo 3: Manuales de uso: 1 semana
  - Tareas:
    - ◊ Guía de uso
    - ◊ Guía de instalación
- Ciclo 4: Wiki en Github: 1 semana
  - Tareas:
    - ◊ Documentación del proyecto
    - ◊ Añadir manuales de uso

Las semanas restantes hasta la primera entrega (26/06/20) se quedan para dar flexibilidad al desarrollo y como margen de previsión ante posibles dificultades imprevistas.

A lo largo del desarrollo se han concertado de forma regular reuniones con el tutor para controlar el progreso del proyecto. Fueron más frecuentes durante la fase inicial de diseño y perfilado y durante las fases finales de documentación y elaboración de la memoria. La periodicidad de las reuniones ha sido de dos semanas aproximadamente y a medida que se acercaba el final del desarrollo se incrementó su frecuencia a revisiones semanales.



# Gamificación y tecnología en la educación

Este capítulo proporciona una visión general sobre la gamificación, algunos métodos y recursos utilizados para su implantación y los resultados obtenidos a través de su aplicación. También explora algunas de las tecnologías actuales que la implementan y las capacidades que ofrecen.

## 2.1. Gamificación

La definición más común de gamificación es “el uso de elementos de diseño de juegos en contextos distintos a los juegos” (Groh, 2012). En los últimos años se ha consolidado como una de las alternativas más atractivas para ofrecer formación desde una aproximación distinta a los métodos tradicionales. Su aplicación no se da únicamente en la educación, sino que también ha captado la atención de las empresas como método de marketing o de instrucción para sus empleados (Korn et al., 2015; Conaway y Garay, 2014).

La gamificación se presenta como una posible solución a uno de los principales problemas que siempre se han dado en la enseñanza, la falta de motivación de los estudiantes. Mediante la aplicación de las mecánicas propias de los juegos podemos potenciar varios aspectos implicados en el proceso de aprendizaje (Lee y Hammer, 2011). La integración de tecnología y recursos online en la educación ha sido objeto de estudio en numerosas ocasiones, obteniéndose generalmente resultados positivos en cuanto al incremento de la satisfacción de los alumnos (Bekele, 2010). Uno de los principales motivos por los cuales los profesores integran gamificación en sus aulas es el entretenimiento y diversión que aportan los juegos. Aprovechando estas características y aplicándolas en las clases se espera conseguir una enseñanza más dinámica y participativa. Sin embargo, para conseguir esto los docentes deben lidiar con la falta de tiempo para preparar recursos y la dificultad para que estos materiales y las actividades sean de utilidad al aprendizaje (Sanchez-Mena y Marti-Parreño, 2016).

## 2.2. Mecánicas de gamificación

Según su origen, la motivación puede clasificarse en *intrínseca*, la voluntad propia del sujeto, o *extrínseca*, aquella debida al entorno y a factores externos (Legault, 2016).

Para potenciar adecuadamente la motivación intrínseca es necesario conocer las necesidades y prioridades del usuario (Richter et al., 2015). La teoría de Maslow propone una pirámide de cinco niveles en los que se clasifican las necesidades que motivan nuestras acciones como individuos. Maslow afirma que deben satisfacerse las necesidades básicas antes de progresar hacia aquellas más complejas. Tomando esta teoría como base, Siang crea su propio modelo para ilustrar las necesidades de los jugadores (Ang y Rao, 2004). Estableciendo el paralelismo entre ambas pirámides, para motivar al usuario habría que proporcionarle en primer lugar suficiente información para comprender el juego. A continuación, el jugador deberá familiarizarse con el juego hasta encontrarse cómodo y sentir que tiene el control. A partir de ese momento intentará superar las pruebas y tareas que se le planteen y, una vez completadas estas, buscará métodos de incrementar sus resultados o estrategias para conseguir un rendimiento más eficiente. Tras esta fase ya se buscará un apartado visual atractivo y por último se explorará el juego intentando buscar los límites del mismo o características ocultas no apreciables para usuarios con poca experiencia.

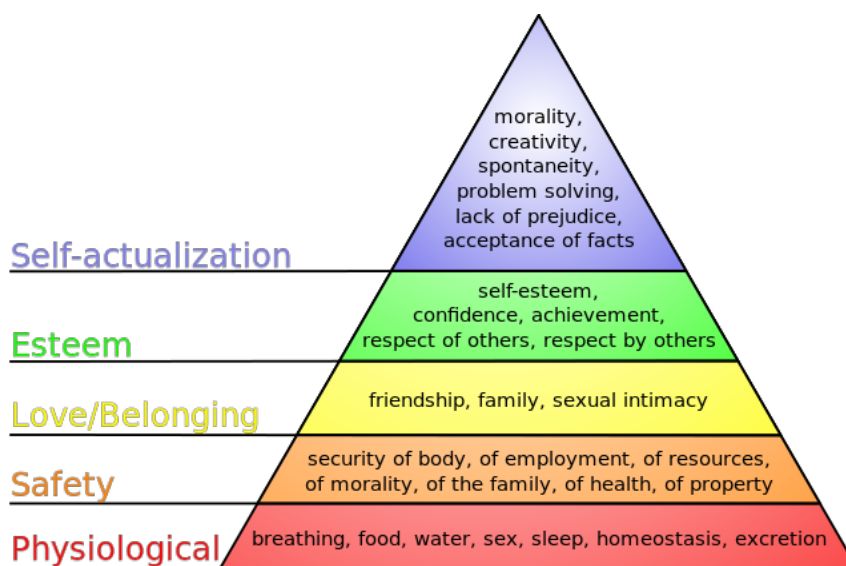


Figura 2.1: Pirámide de las necesidades de Maslow, 1943. Wikimedia Commons (2006)

Como motivación extrínseca en la educación encontramos las notas puestas por los profesores como resultado de la evaluación. Los principales recursos que propone la gamificación como alternativa para fomentar la participación son los logros y puntuaciones (Alsawaier, 2017).

Los sistemas de puntos son una herramienta fundamental por su gran versatilidad y su integración puede satisfacer múltiples propósitos. (Mekler et al., 2013; Zichermann y Cunningham, 2011). La implementación más común del núcleo de estos

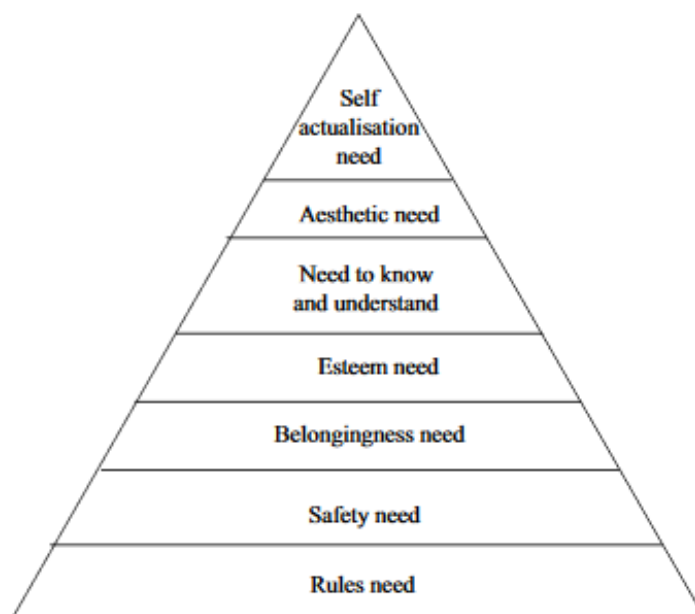


Figura 2.2: Pirámide de las necesidades del jugador. Ang y Rao (2004)

sistemas son los puntos de experiencia, los cuales proporcionan el feedback sobre el progreso del usuario. Otra posible aplicación sería la creación de un sistema virtual de economía en el que pudiesen desbloquearse recompensas a cambio de puntos canjeables. Estrechamente relacionados con las puntuaciones estarían los sistemas de ranking, los cuales muestran el puesto ocupado por un usuario respecto al resto. La efectividad de estas clasificaciones como estímulo de la competitividad depende de la justicia de los criterios considerados para su elaboración (Jia et al., 2017). Los rankings deben tomar en consideración características parametrizables que reflejen fielmente las capacidades de los usuarios. Sin embargo, la utilización de rankings también pueden plantear aspectos negativos, como por ejemplo, la frustración generada por rankings estáticos en los que los mismos usuarios ocupan siempre las mismas posiciones.

Los logros permiten centrar la atención de los usuarios sobre una actividad concreta fijando una meta a cumplir y sirven como reconocimiento de las capacidades. Algunos ejemplos de utilización de logros son el caso de Duolingo, una app gamificada para aprendizaje de idiomas (Solis, 2015), y Khan Academy, una organización educativa sin ánimo de lucro que ofrece lecciones y cursos online (Morrison y Disalvo, 2014). En el caso de Duolingo, al realizar determinadas acciones el usuario recibiría una serie de puntos que le permitirían progresar en metas concretas. Este progreso se manifiesta en la forma de tres niveles de completitud con unos requisitos de puntuación cada vez más exigentes. Análogamente, aunque prescindiendo del sistema de niveles, Khan Academy recompensa a sus usuarios al progresar en sus cursos o por contribuir y colaborar en la elaboración y materiales de la plataforma.

Los anteriores elementos de gamificación dan significado a las tareas asignadas y apelan a la inherente necesidad de realización del usuario, aumentando la motivación a la vez que recompensan el esfuerzo.

### 2.3. Experimentos y proyectos

En los últimos años se han realizado gran cantidad de experimentos para confirmar los beneficios de la aplicación de la tecnología y la gamificación en la educación. En dichos estudios ha participado alumnos de distintos niveles educativos formándose en distintas asignaturas. Gran parte de estas experiencias parten de un diseño experimental basado en la comparación entre dos grupos, cada uno sometido a unas condiciones distintas. Uno de ellos seguiría un plan de estudio tradicional, mientras que en el otro se aplicaría algún tipo de tecnología o gamificación de la enseñanza (Sun-Lin y Chiou, 2019; Kayımbaşoğlu et al., 2016).

Un ejemplo de este tipo de experimentos sería el estudio del impacto de la gamificación sobre las calificaciones obtenidas llevado a cabo en el Instituto Superior Técnico de Lisboa (Barata et al., 2013). Durante tres años se impartió un curso semestral de producción de contenido multimedia y durante los dos siguientes se aplicó gamificación en el mismo, realizándose los ajustes necesarios con el feedback recibido de los alumnos entre el primer y segundo año. Para el estudio se sustituyó la evaluación tradicional por puntos de experiencia y logros coleccionables. Al final del experimento los resultados fueron positivos, la gamificación en el curso repercutió en un gran incremento tanto de la participación como de la proactividad y una mayor satisfacción del alumnado a pesar de una mayor carga de trabajo.

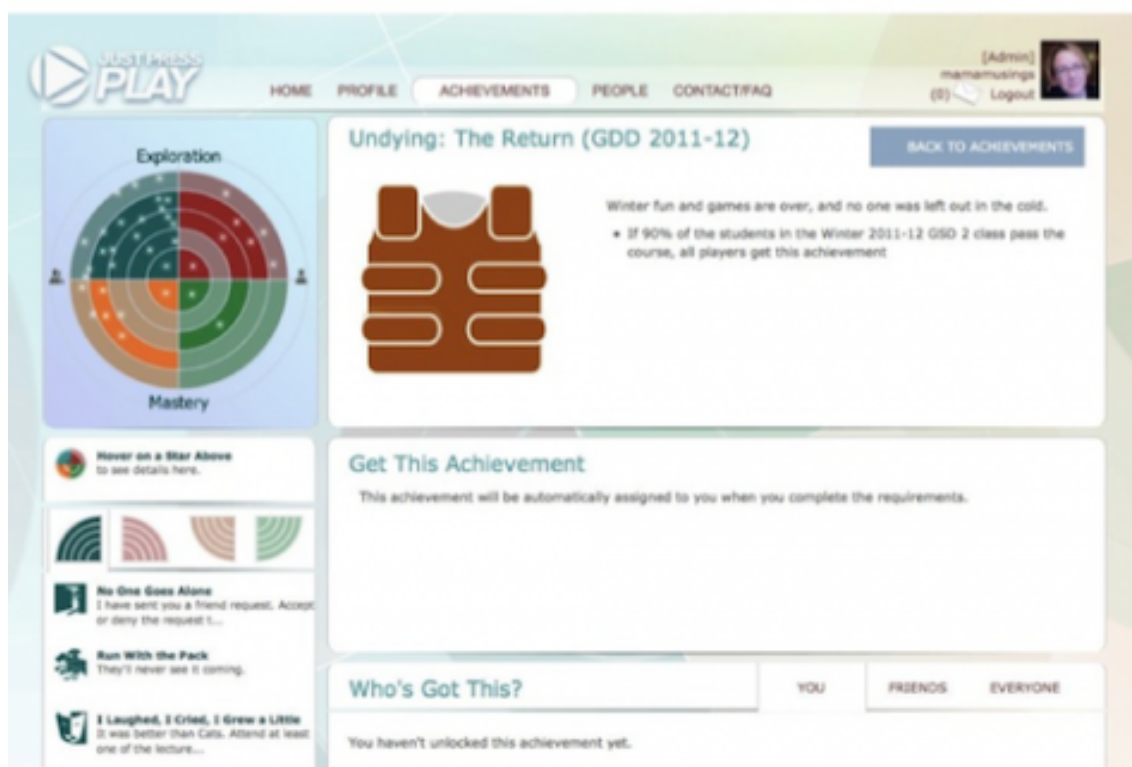


Figura 2.3: Ejemplo de logro en el portal web de Just Press Play. Elizabeth Lawley (2012)

*Just Press Play* es un juego de realidad alternativa desarrollado en el *Rochester Institute of Technology* (RIT) en colaboración con Microsoft (Brinkman, 2011) que intenta llevar el impacto de la gamificación un paso más allá. Ambientando su narrativa en la fundación del RIT, toma como escenario el mundo real para proponer distintas misiones a los usuarios y promover comportamientos tanto académicos como sociales. Los objetivos propuestos son tanto individuales como colectivos y se basan en los principios de dominio y exploración. Para estas últimas los estudiantes dispondrán de un identificador de radio-frecuencia personal que podrán escanear en determinados lugares para registrar su paso. A medida que se completen las misiones se concederán logros. Estos se podrán consultar y comparar con los del resto de la comunidad desde un portal web (Fig. 2.3). La información y datos generados a partir de la participación se destinan al refinamiento de la experiencia para apoyar a los alumnos en sus estudios. El objetivo fundamental de *Just Press Play* es estudiar los desafíos de diseño, privacidad y legales relacionados con la creación de un juego de realidad alternativa en el ámbito de la educación.

## 2.4. Tecnologías

Una de las principales tecnologías que facilitan la labor docente son los LMS (*Learning Management System*). Estos sistemas proporcionan herramientas para gestión de contenido, participación en cursos online y seguimiento del progreso en los mismos (Fig. 2.4). Actualmente los LMS son fundamentales en la infraestructura tecnológica de las instituciones educativas, facilitando la administración por parte de los docentes y el acceso a los recursos académicos por parte de los estudiantes.

Moodle es una de los LMS más utilizados. La plataforma cuenta con más de 25 millones de cursos y 212 millones de usuarios repartidos en 246 países, siendo España el tercer país con más sitios registrados Moodle (2020d). Esta plataforma proporciona a los docentes un sistema con el que crear entornos de aprendizaje personalizados (Moodle, 2018). Su flexibilidad y capacidad de adaptación son algunas de las principales ventajas que han contribuido a su éxito. Distribuida bajo la licencia “GNU General Public License 3” (desde la versión 2.x) ó posterior, y permite la modificación y extensión de la tecnología. Sus herramientas sirven de apoyo tanto al *blended learning* como a cursos online y al tratarse de una plataforma web es accesible desde cualquier tipo de dispositivo y navegador. Moodle incluye incluso sus propios plugin de gamificación (Moodle, 2020b). “Level up” (levelUp, 2019), es uno de los más populares. A través de este los alumnos podrán ganar experiencia al participar en los cursos, subir de nivel, desbloquear nuevos contenidos al progresar y competir por alcanzar el puesto más alto del ranking. Otro ejemplo sería el *plugin* “Game module”, el cuál utiliza varios juegos, como el del ahorcado o las cruzadas, para la resolución de distintos tipos de preguntas: respuesta breve, opción múltiple, verdadero o falso (Moodle, 2020c). Por último, el *plugin* más popular, H5P (H5P, 2020), el cual ayuda en la generación de una amplia variedad de juegos y recursos educativos como pruebas en distintos formatos o vídeos interactivos. Un ejemplo de uso de esta tecnología puede ser un video publicado en YouTube sobre el cuál se han definido una serie de pausas en las que se plantearán al estudiante preguntas que deberá responder.

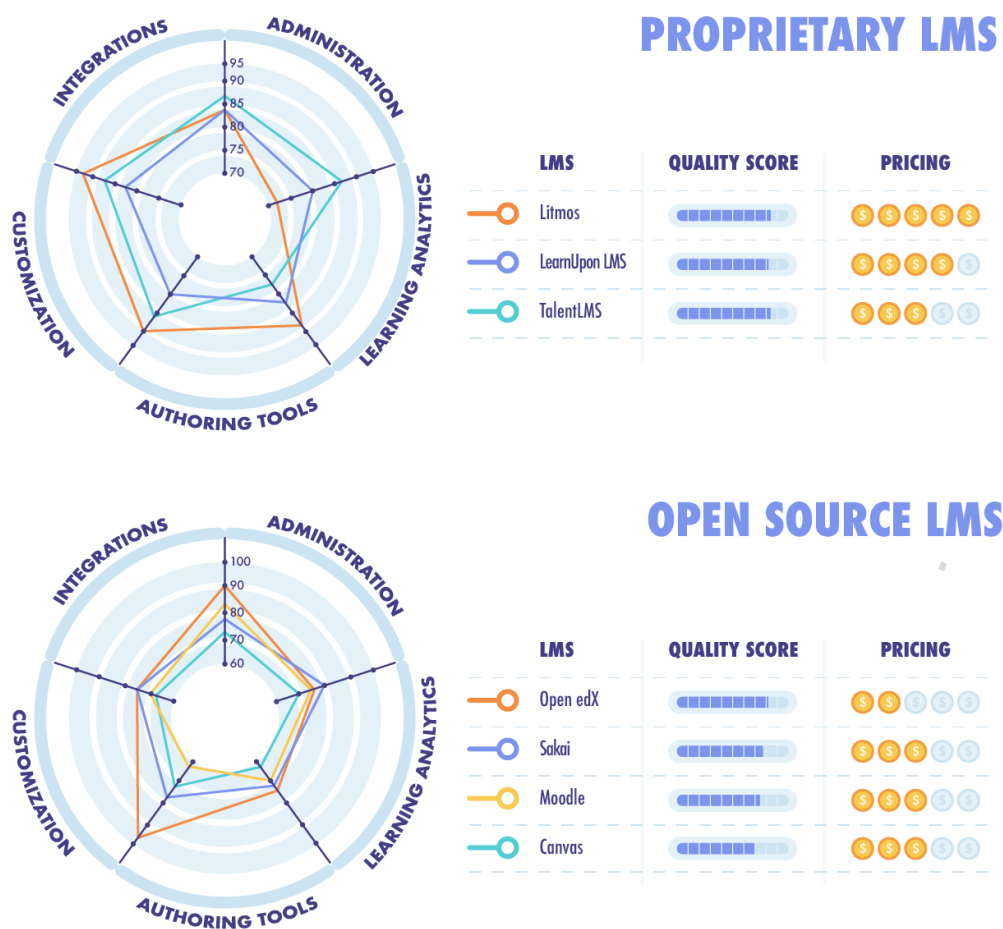


Figura 2.4: Comparación de LMS. Cujba (2020)

Además, Moodle permite la utilización de otras herramientas a través de LTI (Learning Tools Interoperability), una especificación desarrollada por IMS Global Learning Consortium (IMS Global, 2020). Esta define una serie de estándares para la integración de tecnologías externas en otras plataformas como podrían ser los LMS. De este modo se pueden ampliar las actividades disponibles desde Moodle a la vez que se aprovechan las funcionalidades del mismo a través de la herramienta, como podría ser el acceso a listas de alumn.

# Capítulo 3

## Descripción del Trabajo

En este capítulo se presenta el diseño e implementación de la aplicación web.

### 3.1. Especificación de requisitos

#### Caso de uso más común

Un profesor llega a clase. Hoy va a preparar una actividad para sus alumnos. Van a participar en una prueba por grupos y competirán por ver quién consigue más puntos. Accede a la plataforma, carga el fichero con los datos de su clase y junto a los estudiantes organiza los equipos. A continuación, carga el fichero de la prueba, la habilita y reparte las credenciales de acceso. Inician sesión y resuelven la prueba. Cuando todos han terminado el profesor da por finalizada la prueba. Los estudiantes ya tienen acceso a sus resultados y el profesor muestra el ranking de la prueba. los alumnos revisan la clasificación y sus logros para comprobar su progreso. Mientras los estudiantes comparan sus resultados y discuten sobre la prueba el profesor evalúa los resultados de la clase y comprueba las estadísticas.

#### Requisitos

A partir de los objetivos mencionados previamente (ver sección 1.2), y teniendo en cuenta el anterior caso de uso, podemos inferir los siguientes requisitos técnicos para alcanzar una implementación que los satisfaga adecuadamente:

- Interfaz gráfica con una navegación intuitiva y de fácil interacción.
- Almacenamiento centralizado de los recursos.
- Persistencia de la información en una base de datos.
- Distintos perfiles de usuario.

- Acceso mediante contraseña para los profesores.
- Acceso mediante tokens aleatorios para los estudiantes.
- Generación de credenciales de acceso
- Creación de clases a través de ficheros
- Creación de pruebas a través de ficheros
- Organización de los alumnos en grupos.
- Corrección automática de las pruebas
- Generación de los resultados y visualización en tablas.
- Sistema de logros.
- Generación de los rankings.
- Visualización de las estadísticas con las respuestas escogidas para cada pregunta.

## 3.2. Diseño

Analizando los requisitos establecidos y la gran cantidad de componentes a integrar se determinó que el formato más adecuado para el proyecto era una aplicación web. La principal ventaja de esta tecnología es que los usuarios no necesitan realizar ninguna instalación para hacer uso de la plataforma. Esta puede accederse a través de un navegador desde cualquier tipo de dispositivo conectado a internet. Otras características de utilidad son su facilidad de configuración, la centralización de los recursos, su capacidad de almacenamiento mediante la integración de bases de datos y servicios cloud y que se mantienen siempre actualizadas. Como desventajas estarían la necesidad de una conexión a internet y un rendimiento menor que las aplicaciones nativas.

Respecto a la gamificación se barajaron distintas posibilidades, siendo las mecánicas más factibles los sistemas de puntuación y logros y rankings. El concepto entorno al cual estas se organizan es una competición entre equipos de estudiantes que deben resolver pruebas para progresar en un ranking. Un factor fundamental a tener en cuenta es la facilidad del profesorado para interactuar y gestionar el sistema. La opción que se consideró más adecuada en términos de tiempo y esfuerzo para la creación de clases y pruebas fue mediante carga de ficheros.

Una vez definido el proyecto y los requisitos a cumplir llega el momento de organizar el desarrollo. Para comenzar descompondremos el desarrollo tomando como punto de partida los componentes del patrón modelo—vista—controlador típico de las aplicaciones web. Teniendo en cuenta la necesidad de cohesión entre estos tres elementos el objetivo prioritario a cumplir con la selección de las tecnologías es la compatibilidad entre las mismas. A continuación expongo las decisiones tomadas para la implementación de cada componente de la arquitectura.



### 3.2.1. Análisis de tecnologías: vista

Para la construcción de la interfaz gráfica las opciones valoradas son la definición de varias vistas HTML o la utilización de un framework. Los dos principales paradigmas para la construcción de aplicaciones web son el tradicional y el de aplicaciones reactivas. La diferencia fundamental reside en el renderizado de la interfaz. Mientras que las aplicaciones web clásicas se refrescan tras completar una petición, las reactivas utilizan componentes que se actualizan automáticamente. Actualmente las tecnologías de referencia para estas últimas son Angular y React JS, ambas centradas en el prototipado rápido de aplicaciones de página única (Fig. 3.1).

	Angular	React	Vue
# Watchers	3.2k	6.7k	6.2k
# Stars	62k	151k	166k
# Forks	16.9k	29.4k	25.2k
# Contributors	1,129	1,390	293

Figura 3.1: Estadísticas de popularidad de Angular, React y Vue. (Daityari, 2019)

Angular es un framework para aplicaciones web desarrollado en TypeScript y mantenido por Google que se caracteriza por su modelo-vista-controlador basado en una arquitectura de componentes (Angular, 2020). Las interfaces diseñadas con Angular constan de una sola página sobre la que iremos desplegando y sustituyendo distintos elementos. Permite de manera sencilla el desacoplamiento entre la parte visual y la funcional, ya que cada componente se corresponde con un HTML y Javascript, favoreciéndose la reutilización. Sin embargo, esto complica las tareas de testeo y depuración puesto que la detección de los errores se produce en tiempo de ejecución.

React JS es una librería de Javascript para construcción de interfaces de usuario desarrollada por Facebook (React JS, 2020). Al igual que Angular también se basa en componentes con su propia lógica. Aunque no se trate de un framework, y por tanto ofrezca menos herramientas para trabajar, hay que destacar la alta eficiencia y rendimiento de React gracias a su implementación de un DOM (Domain Object Model) virtual. Este actúa como una representación de la página y permite una identificación más rápida de los cambios producidos acelerando la actualización y renderizado de la interfaz.

Las anteriores tecnologías son alternativas válidas y útiles para la construcción del apartado visual rápido y eficiente. Sin embargo, debido a la falta de experiencia previa en el paradigma aplicaciones reactivas es necesario valorar la inversión de tiempo requerida para superar la curva de aprendizaje y empezar a conseguir resultados. Por tanto, y a pesar de ser una solución más artesanal, la opción de crear varias vistas HTML y adoptar un paradigma tradicional resulta más asequible. A pesar de que la cantidad de actualizaciones requeridas por la interfaz será mayor que con un paradig-

ma reactivo, esto no impactará en exceso al rendimiento porque los tiempos de carga iniciales del paradigma tradicional son más rápidos. Quedaría como trabajo futuro la migración de la aplicación a una de las tecnologías anteriormente mencionadas en caso de requerir un mayor rendimiento.

También, en caso de disponer de tiempo suficiente, se estilará la interfaz con Bootstrap, el framework de HTML, CSS y JS más popular para creación de webs responsive (Fig. 3.2). Las principales ventajas de esta tecnología son su amplia adopción, la documentación y recursos disponibles y el soporte para múltiples navegadores. Como contras encontraríamos la inversión temporal en el diseño y programación de la plataforma en caso de querer utilizar elementos distintos a los que Bootstrap ofrece.

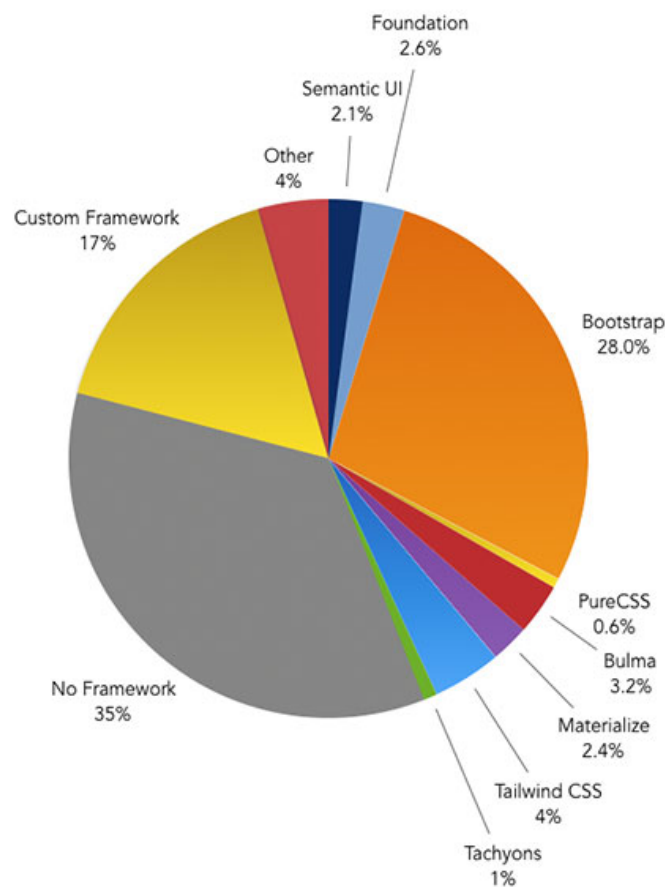


Figura 3.2: Estadísticas de uso de frameworks de CSS. (Watson-Nolan, 2019)

### 3.2.2. Análisis de tecnologías: modelo y controlador

El modelo y el controlador son los pilares entorno a los cuales se implementa la aplicación. Por este motivo, es prioritario que la tecnología seleccionada sea compatible con la mayor cantidad de herramientas posible, agilizando así el proceso de desarrollo. Las tareas fundamentales a resolver serían la configuración de un servidor, la definición del modelo de datos y la gestión e implementación de la lógica del sistema.

El modelo es el responsable de representar la información que será almacenada por el sistema de persistencia, gestionada por el controlador y mostrada en la vista. Será necesario establecer la conexión del sistema con una base de datos. Para el almacenamiento de la información se utilizará una base de datos relacional. Las principales ventajas de esta implementación son la eliminación de elementos duplicados, la seguridad que proporciona y la facilidad para realizar consultas complejas. Su extensibilidad también será de utilidad en previsión de los futuros ajustes introducidos progresivamente sobre el modelo. También será necesaria la conversión y adaptación de la información utilizada por el sistema para su persistencia. Esta tarea es desempeñada por los ORM (*ObjectRelational mapping*), que permiten la creación de bases de datos y la ejecución de consultas usando el paradigma de programación orientada a objetos.

Las tecnologías contempladas para la construcción del modelo y los controladores son Java, PHP y ASP.NET. Java destaca en la creación de aplicaciones *cross-platform*, PHP es una buena alternativa para aplicaciones de bajo presupuesto y .NET es la mejor opción para una implementación basada en tecnologías Microsoft (Fig. 3.3).

PHP es un lenguaje de código abierto especialmente adecuado para desarrollo web ya que permite la integración de código en HTML. La principal diferencia respecto a Javascript, otro de los principales lenguajes para programación web, es que se ejecuta en el servidor y envía al cliente el HTML resultante sin conocerse la lógica implicada en su generación. PHP cuenta con varios frameworks que proporcionan las funcionalidades necesarias para afrontar un desarrollo: conexión con bases de datos, seguridad, implementación de MVC y automatización de tareas comunes (Morris, 2020).

Entre los ORM disponibles para PHP destacan RedBeanPHP y Doctrine (Slant, 2020). RedBeanPHP utiliza una aproximación de no configuraciones por lo que permite prototipado rápido de bases de datos aunque exige una estricta adecuación al formato establecido por la tecnología. Doctrine dispone de gran cantidad de recursos y documentación, una amplia comunidad y similitudes con Hibernate. Sin embargo, requiere de conocimientos de DQL (Doctrine Query Language), el lenguaje de queries propio del ORM, para aprovechar al máximo su potencial y su pesadez puede ser excesiva para proyectos pequeños.

ASP.NET es un framework creado por Microsoft que amplía la base de .NET incorporando herramientas y librerías para la construcción de aplicaciones web. Entre las funcionalidades que ofrece pueden resultar de utilidad la implementación de MVC y la

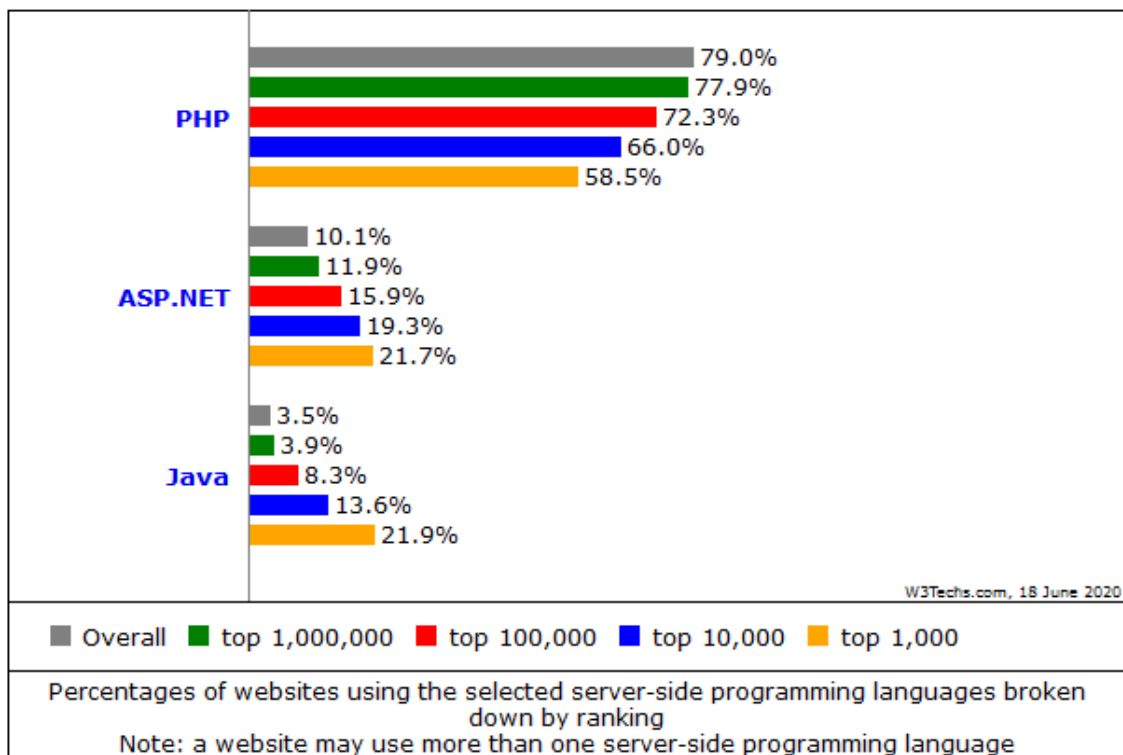


Figura 3.3: Estadísticas de uso de PHP, .NET y Java en aplicaciones web. (W3Techs, 2020)

generación de API's (*Application Programming Interface*) (Microsoft, 2020b,a). Entre las características proporcionadas destacan: templates, un sistema de autenticación, la definición de rutas en el propio código y la posibilidad de integración de múltiples tipos de base de datos.

Para gestionar la persistencia en ASP.NET existen ORM y micro-ORM, los cuales consiguen un mayor eficiencia sacrificando parte de las funcionalidades disponibles. Entre los ORM destacan EF (Entity Framework) y NHibernate y entre los micro-ORM Dapper (Zmaranda et al., 2020). EF reduce los tiempos y costes de desarrollo. Sin embargo, presenta problemas de eficiencia en algunas de las sentencias generadas y es una tecnología complicada de migrar (Entity Framework, 2020b). NHibernate es uno de los ORM más antiguos. Permite flexibilidad en la elección de la base de datos y generación automática de IDs de acuerdo a diferentes estrategias. Sus desventajas son el tiempo inicial para su ejecución y las dificultades para mapear XML. Por último Dapper (Entity Framework, 2020a), presenta un buen rendimiento y eficiencia pero renunciando a la generación de queries y modelos.

Respecto a Java, una de las tecnologías más destacables es Spring (Fig. 3.4). Este es un framework para el desarrollo de software que proporciona una infraestructura completa para el despliegue de aplicaciones en cualquier tipo de plataforma. Spring cuenta con todas las herramientas necesarias para implementar las funcionalidades requeridas por el proyecto; autenticación y control de acceso, soporte para múltiples tecnologías e implementación de MVC (Fig. 3.5). Spring Boot es una extensión de

dicho framework la cual permite la creación de aplicaciones reduciendo la cantidad de configuraciones necesarias.

## What web frameworks do you use, if any?

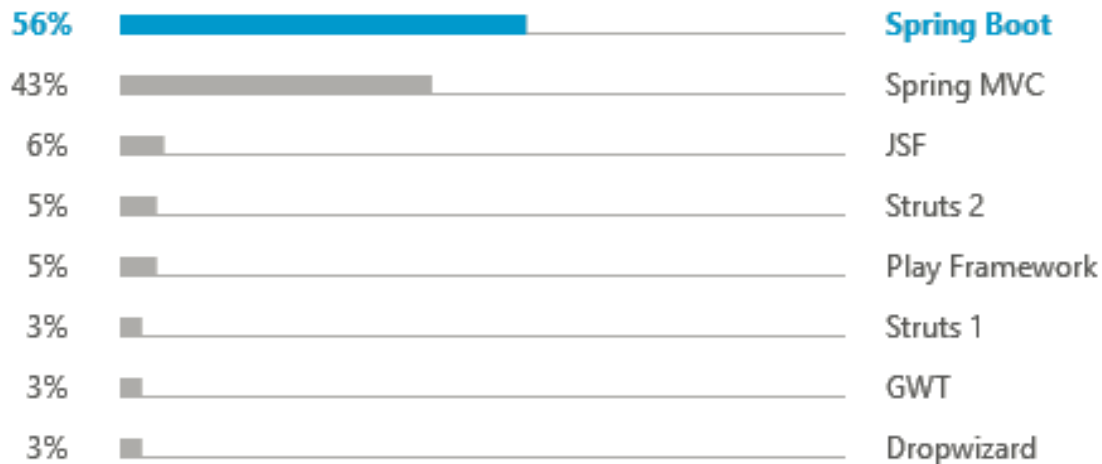


Figura 3.4: Estadística de uso de frameworks en Java. (JetBrains, 2019)

En cuanto a la persistencia, uno de los ORM más utilizados es Hibernate. Este permite el mapeo de clases Java para la generación automática del modelo de datos y sus tablas. No obstante, presenta una ligera curva de aprendizaje al hacer uso de HQL (Hibernate Query Language), su propio lenguaje de consultas. Java también cuenta con JDBC (Java DataBase Connectivity), una interfaz de programación que permite al desarrollador crear sentencias y acceder a una base de datos. La principal desventaja de esta frente a Hibernate es la necesidad de implementar manualmente determinadas tareas, como podría ser el mapeo automático de objetos (DevTeam, 2020).

Una vez analizadas las características de cada posible implementación llega el momento de hacer balance. Ante la falta de experiencia previa en proyectos similares, y dado el tiempo disponible para el desarrollo, sería recomendable contar con una base a partir de la cual empezar a trabajar. A pesar de ser la tecnología más utilizada, PHP queda descartada por la falta de conocimientos del programador. El framework de ASP.NET proporciona una gran variedad de plantillas para la implementación de los elementos fundamentales de una aplicación web y utiliza C# como principal lenguaje de programación. Como alternativa en Java estaría la plantilla de la asignatura “Ingeniería Web” de Manuel Freire (Freire, 2020), la cual construye el esqueleto básico de una aplicación web haciendo uso de las tecnologías de Spring.

Tras el estudio de las distintas opciones para el desarrollo, Java se consolida como la solución más rentable. Trabajando a partir de la plantilla se evitan posibles problemas derivados de la configuración del proyecto. Además, incluye funcionalidades útiles para el sistema como serían la subida y bajada de ficheros de forma segura, la creación de perfiles y la implementación de comunicación mediante *websockets*.

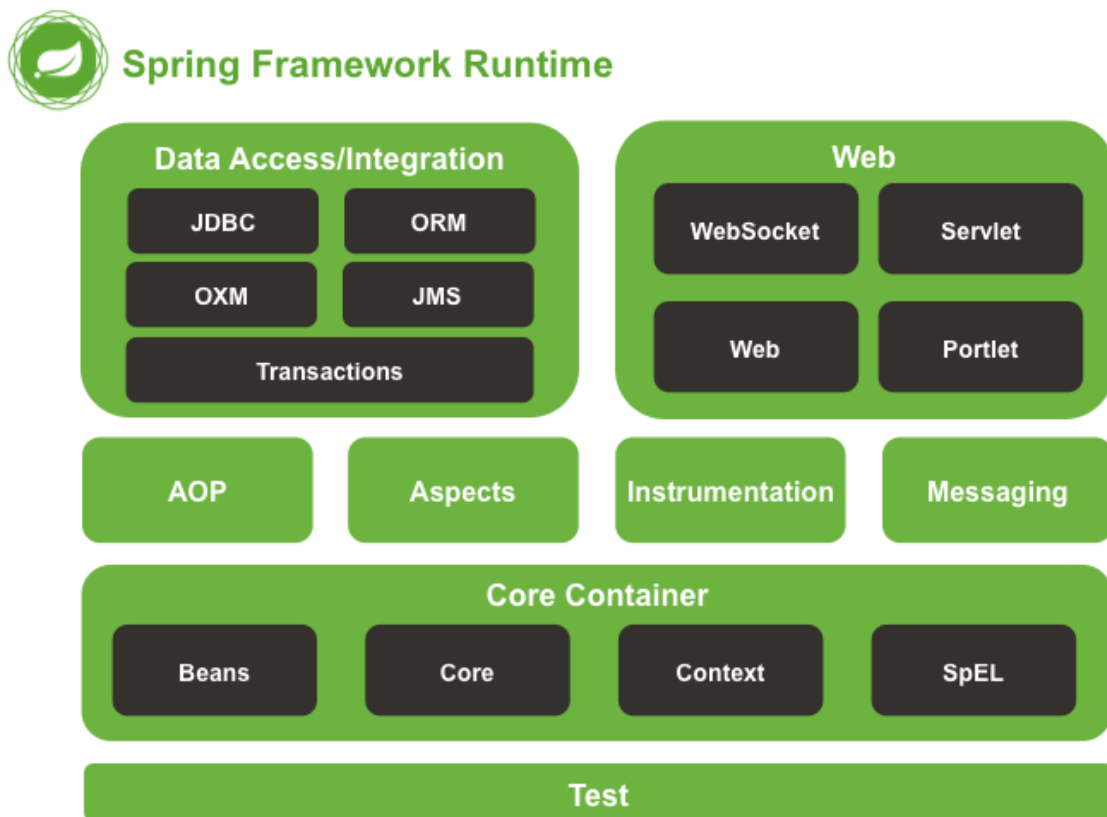


Figura 3.5: Paquetes incluidos en Spring Framework. Spring (2016)

### 3.2.3. Modelo de datos

El dominio de la aplicación está organizado entorno a tres elementos principales: la clase y sus miembros; las pruebas creadas por los profesores; y los logros de alumnos y sus equipos. Las clases constarán de un conjunto de alumnos distribuidos en equipos y un profesor. Las pruebas estarán compuestas de conjuntos de preguntas y cada una de estas incluirá sus correspondientes opciones para responder. Además de las pruebas, se almacenarán los resultados de participación de los estudiantes. Por último, deben distinguirse los logros disponibles para usuarios y equipos, así como definirse las metas que proponen cumplir. A continuación se incluye una explicación más profunda de cada entidad y sus relaciones dentro del modelo (Fig. 3.6):

**User** se utiliza para la creación tanto de profesores como de alumnos, distinguiéndose estos por medio del atributo "roles". El profesorado contará con su propia contraseña, mientras que el acceso de los alumnos se realizará mediante tokens generados por la aplicación. En cuanto a las diferencias entre los tipos de usuario, para cada uno se dará valor a unos atributos específicos. Los alumnos contarán con marcadores para las puntuaciones, tendrán las listas de sus resultados y logros individuales y pertenecerán a un equipo y una clase. Por su parte, los docentes contarán con listas de las clases y pruebas que creen.

**StClass** cada clase tendrá asociado un profesor, una lista de alumnos, una lista de

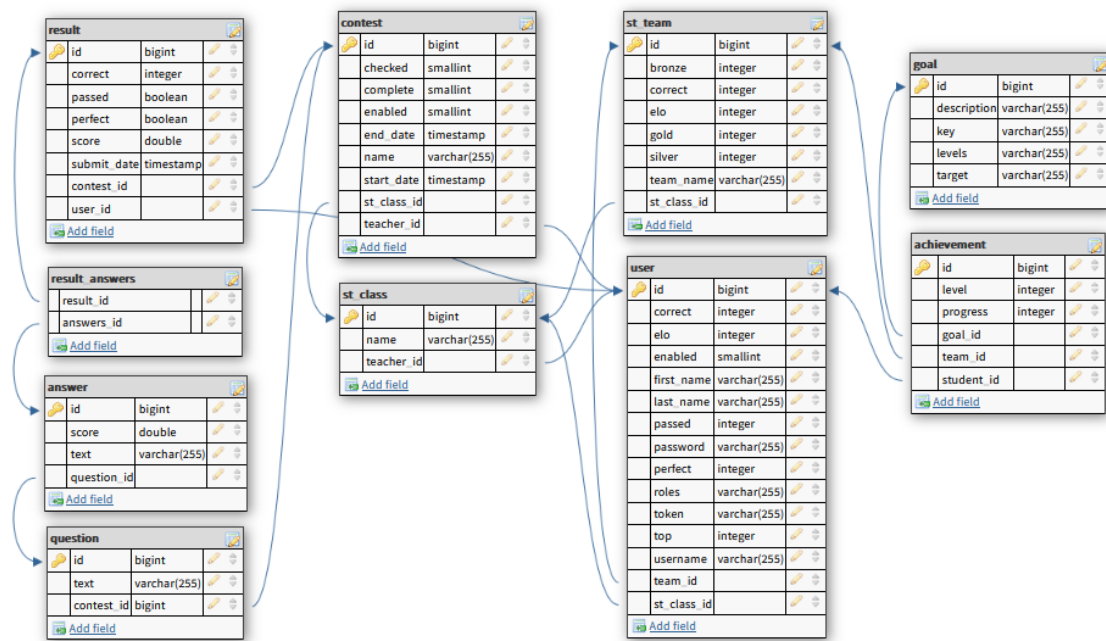


Figura 3.6: Modelo de datos

equipos y una lista con las pruebas asignadas.

**StTeam** cada equipo está asignado a una clase y tendrá como miembros un conjunto de alumnos. Además contará con los correspondientes marcadores de puntuación y una lista de logros colectivos.

**Contest** las pruebas son creadas por los profesores para una clase en concreto y constan de un conjunto de preguntas. Cuentan con tres bytes como atributos para control del estado. El byte "enabled" indica si la prueba está disponible para ser resuelta por los alumnos, el byte "complete" determina el fin de la prueba y el byte "checked" indica si ya se ha realizado la corrección de los resultados asociados. Cuando se habilita y cuando se termina una prueba se fijan las horas de inicio y fin de la misma.

**Question** las preguntas siguen un formato sencillo. Están compuestas por un enunciado y un conjunto de respuestas.

**Answer** las respuestas se asignan a las preguntas y constan de un texto y una puntuación.

**Result** los resultados de los alumnos se asocian a la prueba correspondiente. Una vez resuelta esta se registra la fecha de entrega. Recogen la puntuación obtenida, si ha sido superada y en tal caso, si se ha conseguido sin errores.

**Goal** representa la meta a alcanzar para cumplir un objetivo. Cada una tiene una clave identificativa y se distingue entre aquellas orientadas a méritos individuales y a méritos colectivos. Las metas tienen distintos hitos que indican el nivel de completitud.

**Achievement** actúan como instancias de las metas y recogen el progreso y el nivel alcanzado. Por cada meta se creará un logro para cada estudiante o equipo, comenzando este en el nivel más bajo y con la correspondiente puntuación por defecto.

### 3.2.4. Interfaz

Para favorecer la usabilidad del sistema la interfaz gráfica debe ser lo más simple e intuitiva posible. Cada funcionalidad de la plataforma estará asociada con su correspondiente vista. Estas constan de tres elementos: una barra superior para navegación, el cuerpo y el footer acreditando el uso de la plantilla.

En los primeros bocetos se planteó la navegación por medio de un menú lateral (Fig. 3.7). Sin embargo, y para optimizar la utilización del espacio se migró esta a la zona superior de la vista. Inicialmente la barra de navegación se utilizará para el inicio de sesión y según el usuario loggeado sea un estudiante o un profesor incluirá los botones para acceder a las funcionalidades disponibles para ese rol.

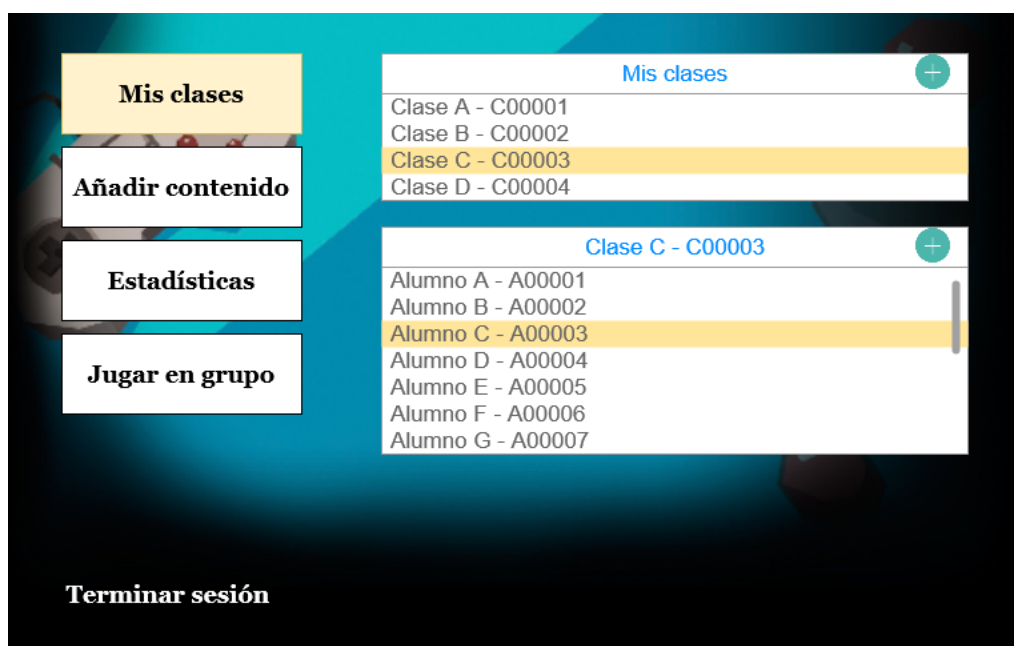


Figura 3.7: Ejemplo de boceto inicial con navegación mediante menú lateral

Las funcionalidades comunes para ambos tipos de usuario serían los rankings y la resolución de pruebas (Fig. 3.8). Los estudiantes podrán ver el ranking global de la clase y las clasificaciones para cada prueba finalizada. Los profesores visualizarán los rankings globales de cada clase que hayan creado mientras que si quisiesen ver la clasificaciones por prueba, para evitar la saturación, deberán hacerlo desde la vista de pruebas.

Respecto a la resolución de pruebas, los profesores podrán probar cualquiera que hayan creado mientras que los estudiantes podrán participar en aquellas disponibles para su clase. Para responder las preguntas el usuario deberá seleccionar la opción



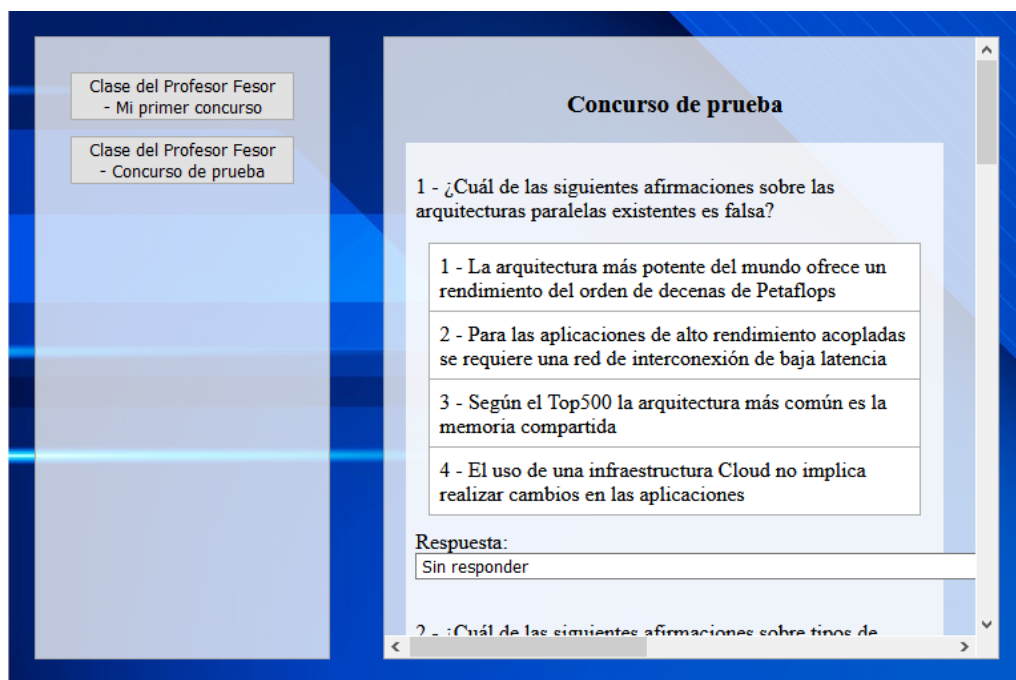


Figura 3.8: Ejemplo de funcionalidad común. Resolución de pruebas

que considere correcta entre las disponibles en las listas desplegadas. Los docentes verán inmediatamente los resultados y tendrá la opción de reintentar la prueba, mientras que los alumnos deberán esperar a que el profesor finalice la prueba para ver sus resultados.

Por defecto, la vista de clases mostrará aquellas creadas por el correspondiente profesor. Al seleccionar una en concreto se mostrará la lista de alumnos, los equipos en los que se han organizado y las pruebas asignadas. Desde este último apartado podrán crearse y asignarse pruebas mediante la subida de un fichero. Este mismo método se utiliza para la creación de las clases. Una vez procesado el fichero la vista mostrará un editor con los alumnos cargados y un panel a través del cual generar los grupos. Para su organización, el profesor simplemente deberá arrastrar los alumnos a los contenedores de los equipos y guardar los cambios.

En la vista de pruebas se recogen aquellas creadas por el docente. Al seleccionar una se mostrarán las preguntas y respuestas correspondientes, así como el estado de la prueba. Desde el panel de gestión esta se podrá habilitar, deshabilitar o finalizar. A medida que los alumnos la vayan resolviendo se incluirán los resultados obtenidos en una tabla. Además, en base a estos se generarán las gráficas con las respuestas dadas por los estudiantes para comprobar más fácilmente el rendimiento de la clase en cada pregunta.

Por último están las vistas de perfil y de equipo, exclusivas para los estudiantes. Desde el perfil puede cambiarse la imagen y nombre de usuario y comprobar el progreso de los logros. Análogamente está la vista de equipos, que muestra los integrantes del grupo, así como las copas conseguidas.

### 3.2.5. Inicio de sesión

Para el inicio de sesión se implementarán dos modalidades: a través de una contraseña y a través de un token. Los profesores contarán con una contraseña propia que permitirá el acceso a la plataforma en cualquier momento. Para añadir una capa de encriptado las contraseñas se codifican haciendo uso del paquete de seguridad proporcionado por Spring Framework. Los tokens se utilizan para el acceso de los estudiantes y actúan a modo de contraseña. Los alumnos no sabrán cuál es su token sino que se les distribuirá un código QR que contiene una URL única para iniciar sesión y acceder directamente a la plataforma. De este modo se pretende dificultar el tráfico de cuentas para conseguir una competición justa y la validez del sistema como método de evaluación.

La vista de las clases incluye un botón para la generación de los códigos QR. A través de este el profesor podrá descargar un documento en formato PDF con las credenciales de acceso de sus alumnos. Cada vez que se descargue uno de estos documentos se creará un nuevo token para cada estudiante, permitiéndose así el control del acceso a la plataforma. La generación de los tokens se realiza de manera aleatoria y consisten en una cadena de caracteres alfanuméricos de una longitud fija.

### 3.2.6. Carga de datos

Para la creación de las clases y pruebas se consideró que la subida de ficheros era la solución más sencilla para resolver el problema. El formato de estos debe ser lo más simple posible para facilitar el trabajo de los docentes. Puesto que el desarrollo del proyecto se realiza en Java se optó por ficheros JSON para cargar la información.

Los ficheros para clases contienen el nombre de la clase y una lista con los nombre y apellidos de los estudiantes. El resto de datos se generarán por defecto o a partir del uso de la aplicación.

```
{
  "nombreClase": "Clase de ejemplo",
  "alumnos": [
    {
      "nombre": "Gloria",
      "apellidos": "Cebrian Pereiro"
    },
    {
      "nombre": "Agustin",
      "apellidos": "Grande Borrego"
    },
    {
      "nombre": "Maria",
      "apellidos": "Burgos Acunia"
    }
  ]
}
```

}

Listado 3.1: Formato de archivo con alumnos

Para que la edición de las pruebas por parte de los docentes fuese lo más sencilla posible se ha tomado como ejemplo el formato Aiken (Moodle, 2020a), el cuál destaca por su legibilidad. Este se ha adaptado para añadir diferentes grados de acierto en las respuestas. En los ficheros se especificará un nombre y una lista de preguntas.

Cada pregunta está compuesta de un enunciado y un conjunto de posibles respuestas acompañadas de su valor. Este valor no actúa como una puntuación absoluta sino que representan un porcentaje, pudiendo así distinguir el grado de acierto de las respuesta. Teniendo esto en cuenta pueden darse opciones correctas (valor 1), opciones erróneas (valor 0 o negativo si se desea aplicar penalización) y opciones válidas aunque solo parcialmente (valores intermedios entre 0 y 1).

```
{
  "nombreConcurso": "Concurso de prueba",
  "preguntas": [
    {
      "enunciado": "Cual de las siguientes afirmaciones sobre las
                    arquitecturas paralelas existentes es falsa",
      "respuestas": [
        "La arquitectura ms potente del mundo ofrece un rendimiento
          del orden de decenas de Petaflops-->0",
        "Para las aplicaciones de alto rendimiento acopladas se
          requiere una red de interconexion de baja latencia-->0",
        "Segun el Top500 la arquitectura ms comun es la memoria
          compartida-->1",
        "El uso de una infraestructura Cloud no implica realizar
          cambios en las aplicaciones-->0"
      ]
    },
    {
      "enunciado": "Cual de las siguientes afirmaciones sobre tipos de
                    aplicaciones es falsa?",
      "respuestas": [
        "Las aplicaciones de alta productividad no requieren el uso
          de un modelo de programacion paralela-->0",
        "El modelo de alta productividad es compatible con el de alto
          rendimiento ya que cada tarea puede requerir el uso
          simultaneo de varios procesadores-->0",
        "Los flujos de trabajo (workflows) son un ejemplo de
          aplicacion de alta productividad-->0",
        "Las aplicaciones de alta productividad son solo aquellas que
          se ejecutan sobre arquitecturas de ejecucin
          oportunistas-->1"
      ]
    }
  ]
}
```

```
}  
]  
}
```

Listado 3.2: Formato de archivo con preguntas para un concurso

### 3.3. Implementación

La mayor parte de la lógica se concentra en los dos controladores, uno para las funcionalidades dedicadas a los docentes y otro para los estudiantes. Para facilitar su desarrollo y pruebas estos han sido descompuestos a su vez en múltiples controladores, especializados cada uno en una vista (Fig. 3.9).

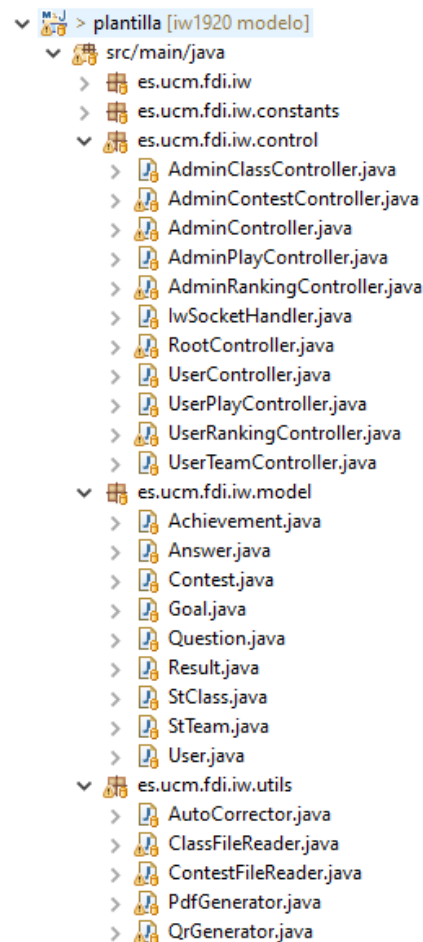


Figura 3.9: Controladores, modelo de datos y herramientas del sistema

El siguiente diagrama (fig. 3.10) muestra un posible flujo para una ejecución completa de la plataforma en la que se llevan a cabo todos los pasos del proceso de organización de una actividad.

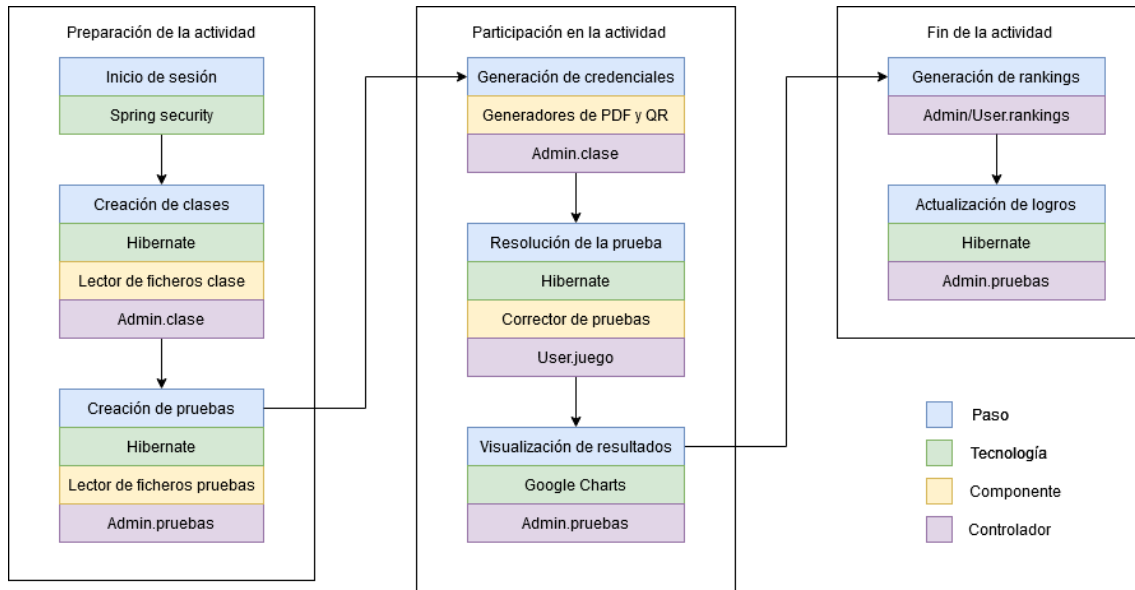


Figura 3.10: Flujo completo de ejecución de la plataforma

### 3.3.1. Inicio de sesión

Como ya se comentó en el apartado anterior de diseño existen dos modalidades de inicio de sesión 3.2.5. En el caso de los profesores, la contraseña se definirá en el momento de creación del perfil. La contraseña en el caso de los estudiantes se corresponderá con un token aleatorio, el cual cambiará cada vez que el profesor genere nuevas credenciales de acceso. Las contraseñas se codificarán haciendo uso de Spring Security para añadir una capa de criptografía.

Los tokens consistirán en una cadena alfanumérica de una longitud determinada. Para evitar cadenas confusas se prescindirá de caracteres similares como podrían ser la “l” y la “I” o el “0” y la “O”. (List. 3.3).

```

public void createAndSetRandomToken(final int length) {
    final char[] chars =
        "ABCDEFGHJKLMNPQabcdefghijkmnopq123456789".toCharArray();
    final StringBuilder token = new StringBuilder();
    final Random r = new Random();
    for (int i = 0; i < length; i++) {
        token.append(chars[r.nextInt(chars.length)]);
    }
    this.token = token.toString();
}
  
```

Listado 3.3: Generación de un token aleatorio

Para gestionar el acceso de los estudiantes a la plataforma el profesor podrá cambiar las credenciales a través del botón “Descargar QR de acceso” (Fig. 3.11) en la vista de la clase . Se generará un documento PDF con un código QR por estudiante, el cual permitirá el inicio de sesión mediante token 3.5. Para esto se utiliza Zxing, una

librería open source de procesamiento de imágenes de código de barras mantenida por Google. Para la construcción del documento PDF se utilizan las funcionalidades open source de iText PDF, un toolkit para generación y manipulación de este tipo de ficheros. Una vez completado el proceso el documento se almacenará en el sistema de ficheros compartidos de la plataforma para permitir su descarga.

---

```
public static void generateQrCode(User user) throws UnknownHostException {
    InetAddress inetAddress = InetAddress.getLocalHost();
    String url = "http://" + inetAddress.getHostAddress() + ":"
        + ConstantsPdfFile.PORT + "/token/" + user.getToken();
```

---

Listado 3.4: Contenido del código QR



José, Abril Riesco	ST.00A	
Alicia, Valls Arce	ST.00D	

Figura 3.11: Detalle de un fichero con las credenciales de acceso

---

```
public StreamingResponseBody getQrFile(@PathVariable("id") long id,
    @PathVariable("classId") long classId,
    Model model, HttpSession session) throws IOException, DocumentException
{
    StClass stClass = entityManager.find(StClass.class, classId);
    if (stClass.getStudents() == null || stClass.getStudents().isEmpty()){
        log.info("Error al acceder a los datos de los alumnos");
    } else {
        log.info("Generando tokens para alumnos");
        for (User u : stClass.getStudents()) {
            u.createAndSetRandomToken(Constants.TOKEN_LENGTH);
            u.setPassword(passwordEncoder.encode(u.getToken()));
        }
        log.info("Creando fichero QR de la clase");
        String qrFile = PdfGenerator.generateQrClassFile(stClass.getStudents(),
            stClass);
        uploadToTemp(qrFile, Long.toString(classId));
    }

    File f = localData.getFile("qrcodes\\" + Long.toString(classId) ,
        ConstantsPdfFile.QR_FILE + "." + ConstantsPdfFile.PDF);
    InputStream in = new BufferedInputStream(new FileInputStream(f));
    return new StreamingResponseBody() {
        @Override
        public void writeTo(OutputStream os) throws IOException {
            FileCopyUtils.copy(in, os);
        }
    };
}
```

```
}  
};  
}
```

Listado 3.5: Generación del documento PDF con las credenciales de acceso

### 3.3.2. Creación de clases

Como ya se ha comentado anteriormente la creación de las clases se realiza a través de ficheros JSON, realizándose la subida de estos desde la vista de clases. Por cada fichero cargado se creará una nueva clase con el nombre indicado, vinculada al correspondiente profesor, y tantos usuarios como estudiantes se hayan especificado.

Para cada alumno se generará por defecto un nombre de usuario y contraseña a los que se asignará el valor del token. Teniendo en cuenta la creación de múltiples clases, a través de esta implementación se pretende evitar la duplicidad de nombres de usuarios en la base de datos y los consiguientes problemas. También se inicializarán los marcadores con los valores mínimos y se creará un logro por cada meta disponible en el sistema y usuario.

Una vez generada la clase será posible editar los equipos. La vista mostrará la lista de alumnos y un panel en el cual se crearán tantos contenedores como se especifique (Fig. 3.12). Para la formación de los grupos bastará con pinchar los nombres de los estudiantes y arrastrarlos hasta el equipo donde vaya a asignarse. Al guardar, un código Javascript procesará el estado de la vista y enviará la distribución al controlador, el cual generará los equipos, creará los logros para cada uno y asignará cada estudiante al grupo correspondiente.



Figura 3.12: Creación de los equipos a través de la vista

### 3.3.3. Creación de pruebas

Las pruebas están asignadas a una clase en concreto. Para la creación de una nueva se debe cargar el fichero desde el correspondiente apartado de la vista de una clase. Tras ser procesado se crearán entidades para la prueba, sus preguntas y las respuestas. Además por cada pregunta se generará una respuesta adicional de puntuación nula que se correspondería con la opción de no responder.

Cada prueba tiene asociada un lista en la que se irán almacenando los resultados de los estudiantes a medida que sea resuelta. En función de estos se generarán las gráficas que ilustran las estadísticas de respuesta para cada pregunta, permitiendo a los docentes realizar el seguimiento del rendimiento de la clase de manera sencilla (Fig. 3.13).

Para la elaboración de dichas gráficas se ha hecho uso de la API de Google Charts. En la carga de la vista de la prueba se procesarán las estadísticas generadas por el controlador, accesibles a través del modelo. Estas consisten en una lista de strings, uno por pregunta, los cuales recogen el enunciado y cada una de las opciones para responder junto al número de veces que han sido seleccionadas (List. 3.6). Posteriormente, un código javascript procesa dicha información, extrae cada campo y lo adapta al formato requerido por la API (List. 3.7).

---

Enunciado | Respuesta1 --> N1 ; Respuesta2 --> N2 ; ...

---

Listado 3.6: Formato de las estadística de una pregunta

---

```
for(i=0; i < statList.length; i++) {
  chartData = [];
  chartData.push(["Respuesta", "Estudiantes"]);
  questionStats = statList[i].split("|");
  answerStats = questionStats[1].split(";");

  for(j=0; j < answerStats.length; j++) {
    answer = answerStats[j].split("-->");
    chartData.push([answer[0], parseInt(answer[1])]);
  }

  console.log(chartData);

  data = google.visualization.arrayToDataTable(chartData);
  options = {'title':questionStats[0], 'sliceVisibilityThreshold':0};
  chart = new google.visualization.PieChart(document.getElementById('g' +
    i));
  chart.draw(data, options);
}
```

---

Listado 3.7: Creación de las gráficas con las estadísticas de cad pregunta





Figura 3.13: Resultados y estadísticas de una prueba

### 3.3.4. Participación en pruebas

Los docentes podrán testear las pruebas creadas indefinidamente. Los alumnos tendrán acceso solamente a las pruebas asignadas a su clase y podrán resolver estas una única vez mientras estén disponibles. Para la resolución simplemente se debe seleccionar una opción de la lista desplegable por cada pregunta. Al terminar se procede a la corrección inmediata, aunque los alumnos solo tendrán acceso a sus resultados una vez el profesor haya finalizado la prueba.

Para cada respuesta se revisará si es correcta, aquellas con una puntuación asignada de 1, y se calcularán los puntos conseguidos. Si la suma de estos es igual o superior a la puntuación mínima necesaria la prueba se dará por aprobada. Si además todas preguntas han sido respondidas correctamente se registrará el resultado como perfecto. En base a la puntuación y a la superación de la prueba se actualizarán los correspondientes logros, incrementando los marcadores del usuario y añadiendo la aportación individual al progreso del grupo (List. 3.8).

```
public static Result correction(User user, StTeam team, Contest contest,
    List<String> answerList) {
    Result result;
```

```
int correct = 0;
double totalScore = 0;
boolean passed = false;
boolean perfect = false;

Answer answer;
int index;
double score;

result = new Result();
result.setContest(contest);
result.setUser(user);
result.setAnswers(new ArrayList<>());
result.setSubmitDate(new Date());

for (int i = 0; i < answerList.size(); i++) {
    index = Integer.valueOf(answerList.get(i));
    answer = contest.getQuestions().get(i).getAnswers().get(index);
    result.getAnswers().add(answer);

    score = answer.getScore();
    totalScore += score * Constants.Q_SCORE;
    if (score == 1) {
        correct++;
    }
}

if (totalScore >= answerList.size() * Constants.Q_SCORE *
    Constants.SCORE_LIMIT) {
    passed = true;
    if (totalScore >= answerList.size() * Constants.Q_SCORE) {
        perfect = true;
    }
}

result.setCorrect(correct);
result.setPassed(passed);
result.setPerfect(perfect);
result.setScore(Math.round(totalScore));

if (user.getRoles().equals(Constants.USER_ROLE)) {
    user.setElo(user.getElo() + (int)Math.round(totalScore));
    user.setCorrect(user.getCorrect() + correct);
    user.setPassed(user.getPassed() + 1);
    user.setPerfect(user.getPerfect() + 1);
    team.setElo(team.getElo() + (int)Math.round(totalScore));
    team.setCorrect(team.getCorrect() + correct);
}

return result;
}
```

Listado 3.8: Corrección de las respuestas dadas para una prueba

### 3.3.5. Generación de rankings

Dentro de la aplicación existen dos tipos de rankings, uno global (Fig. 3.14) que se va actualizando con cada prueba y muestra el progreso de los estudiantes y equipos y los rankings con la clasificación asociada a cada prueba. Para fomentar la participación y evitar la desmotivación de la clase por el monopolio de las primeras posiciones se ha ajustado la lógica de generación de los rankings de modo que aquellos estudiantes con la misma puntuación compartan la misma posición.



Figura 3.14: Rankings global de estudiantes y equipos

Para la generación del ranking global simplemente se toman la lista de alumnos y equipos de una clase y se ordenan de acuerdo a la puntuación acumulada. La generación del ranking de cada prueba es algo más compleja ya que se realiza a partir de los resultados. Para la clasificación de los equipos se creará un mapa con las puntuaciones de cada uno, el cual se irá actualizando con las a medida que se determina la posición ocupada por cada estudiante (List. 3.9). Para conseguir esta implementación y poder usar un equipo como clave del mapa es necesaria la modificación del hash en la clase `StTeam`. Obtenidas las puntuaciones colectivas se ordenará el mapa y se aplicará la lógica de posicionamiento al igual que se haría en el ranking global.

```
private void getContestRanking(List<StTeam> teams, List<Result> results,
    Contest contest, Model model) {

    Map<StTeam, Double> sumScores = new HashMap<>();
    for (StTeam s : teams) {
        sumScores.put(s, 0.0);
    }

    List<Integer> positionUser = new ArrayList<>();
    StTeam team;
    int pos = 1;
    double score;
    double max;
    if (results != null && !results.isEmpty()) {
```

---

```

max = results.get(0).getScore();
//En caso de que varios estudiantes hayan logrado la misma puntuación
//compartir la posición de la clasificación
for (int i=0; i < results.size(); i++) {
    score = results.get(i).getScore();

    if ( score < max) {
        pos++; max = score;
        positionUser.add(pos);
    } else {
        positionUser.add(pos);
    }

    team = results.get(i).getUser().getTeam();
    if (sumScores.keySet().contains(team))
        sumScores.put(team, sumScores.get(team) + score);
    else
        sumScores.put(team, score);
}

model.addAttribute("positionUser", positionUser);
if (contest.getChecked() == 0) {
    updateAchievementsUser(results, positionUser);
}
}

```

---

Listado 3.9: Asignación de posiciones en el ranking asociado a una prueba

### 3.3.6. Logros y progresión

Los logros recompensan tanto el esfuerzo individual como colectivo de los estudiantes (Fig. 3.15). La actualización del progreso se realiza en dos momentos concretos. Cuando un alumno completa una prueba sus logros se ajustan de acuerdo a los resultados obtenidos, los cuales se suman al progreso acumulado del equipo. Cuando el profesor pone fin a una prueba se genera el ranking con la clasificación definitiva y se procede al reparto de recompensas según la posición alcanzada. En el caso de los equipos además, en cada prueba se otorgará un trofeo de oro, plata o bronce a aquellos equipos que terminen entre las 3 primeras posiciones, los cuales lucirán en el perfil grupal.

Los logros cuenta con 5 hitos, siendo cada vez más exigentes los requisitos necesarios para aumentar el nivel. Cuando el progreso de un logro se actualiza se compara con el hito correspondiente a ese nivel y meta y en caso de haber alcanzado la marca se incrementa (List. 3.10). Cada logro está asociado a un factor diferente, por lo que para su distinción en el momento de la actualización cada meta cuenta con su propia clave identificativa.

---

```

public static List<Achievement> updateAchievementsUser(List<Achievement>
    achievements, User user) {
    String[] levels;

```

```

for (Achievement a: achievements) {
    levels = a.getGoal().getLevels().split(",");

    switch(a.getGoal().getKey()) {
        case("CORRECT"):
            a.setProgress(user.getCorrect());
            if (user.getCorrect() >= Integer.parseInt(levels[a.getLevel()])) {
                a.setLevel(a.getLevel() + 1);
            }
            break;
        case("PASSED"):
            a.setProgress(user.getPassed());
            if (user.getPassed() >= Integer.parseInt(levels[a.getLevel()])) {
                a.setLevel(a.getLevel() + 1);
            }
            break;
        case("ELO"):
            a.setProgress(user.getElo());
            if (user.getElo() >= Integer.parseInt(levels[a.getLevel()])) {
                a.setLevel(a.getLevel() + 1);
            }
            break;
        default:
            break;
    }
}

return achievements;
}

```

---

Listado 3.10: Actualización de los logros de un alumno tras la resolución de una prueba

## 3.4. Pruebas y testeo de la aplicación

Esta sección describe varios casos de uso usados para verificar que la ejecución de las funcionalidades principales del sistema producen los resultados especificados. Para cada prueba se especifican los pasos seguidos, el resultados esperado en cada uno y los errores que pueden producirse.

### 3.4.1. Caso de uso: creación de una clase

- Resultado: se ha añadido una nueva clase a las creadas por el profesor. Es posible acceder a los perfiles de los estudiantes creados. Los equipos se han formado correctamente.
- Pasos a seguir:

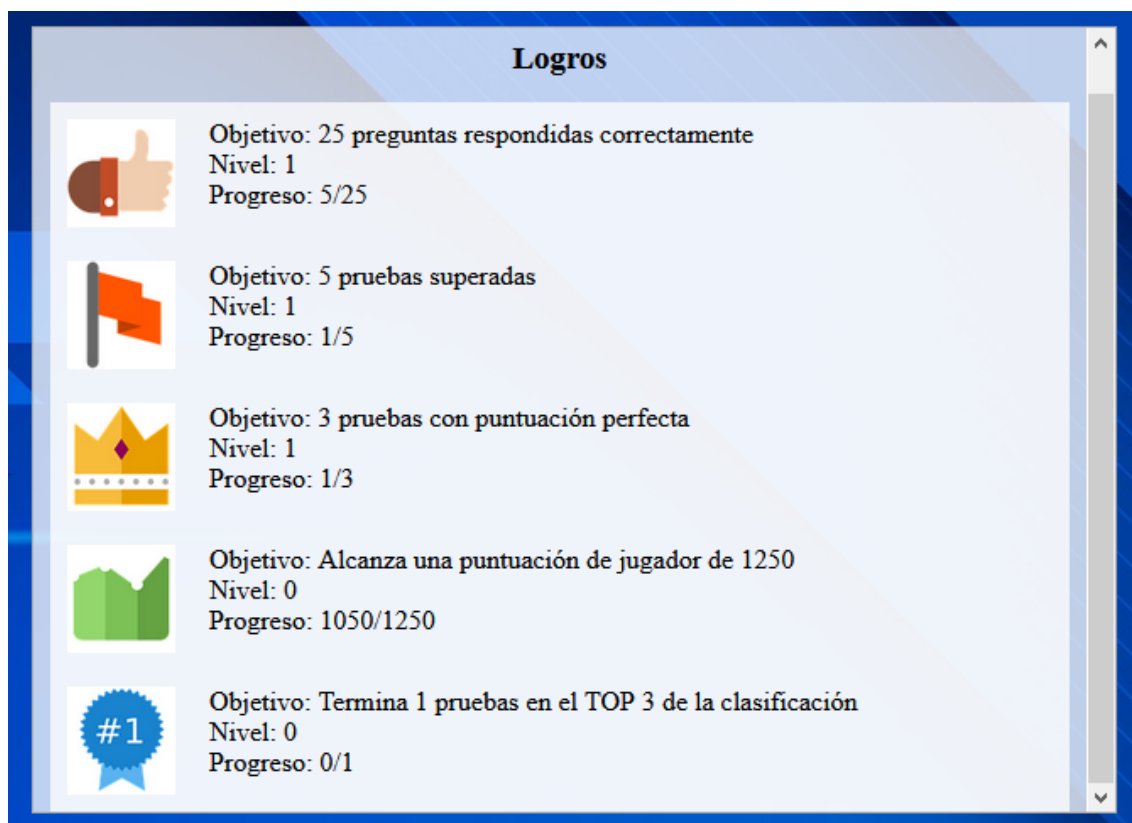


Figura 3.15: Lista de logros de un alumno y su progreso

- Selección y carga del fichero: en el profesor cargará su fichero desde la vista de clases.
  - Resultado: se crea una clase con sus alumnos y se persisten en base de datos. Se avanza a la pantalla de edición de equipos.
- Creación de los equipos: el profesor indicará un número de equipos y organizará a los estudiantes en los correspondientes contenedores.
  - Resultado: en base a la distribución de los alumnos se crean los equipos y se persisten en base de datos. Se avanza a la vista de la clase creada.
- Generación de las credenciales de acceso: el profesor descargará el documento PDF con los QR para acceso mediante token.
  - Resultado: se genera un token aleatorio por alumno. Se genera un documento PDF con la lista de alumnos y sus respectivos QR de acceso. El documento se añade al sistema de ficheros compartidos y es descargable.
- Inicio de sesión de alumnos: el alumno escanea el código QR para acceder a la plataforma.
  - Resultado: se inicia sesión automáticamente. Se avanza a la vista de usuario. Existe una lista de logros aún sin progreso. En la vista de equipo se muestran los miembros del grupo, los logros grupales y los trofeos.

■ Errores:

- Fichero erróneo: el formato de fichero no es el correcto o presenta errores.
  - Resultado: se crea una clase vacía.
- Interrupción de la creación de equipos: se abandona el proceso o se guardan los cambios sin haber organizado equipos.
  - Resultado: la clase y alumnos se crean correctamente pero no existen equipos asignados.
- Ranking global sin equipos: se navega al ranking de una clase para la que no se han creado y persistido en base de datos los equipos.
  - Resultado: se muestra una pantalla de error.
- No se puede crear el fichero: se intenta obtener las credenciales de una clase errónea.
  - Resultado: se muestra una pantalla de error.
- Nombre de usuario duplicado: en caso de que existan en base de datos dos usuarios con el mismo nombre de usuario se producirá una colisión.
  - Resultado: se muestra un mensaje de credenciales erróneas y no permite el acceso.
- Cantidad de equipos no corresponde: en caso de que la cantidad especificada en el input sea menor que el número de contenedores se crearán solamente la cantidad indicada. Si el input es mayor que el número de contenedores durante el procesado se lanzará una excepción.
  - Resultado: se muestra una pantalla de error.

### 3.4.2. Caso de uso: creación de una prueba

- Resultado: se ha añadido una nueva prueba a la correspondiente clase.
- Pasos a seguir:
  - Selección y carga del fichero: en el profesor cargará su fichero desde la vista de una clase concreta.
    - Resultado: se crea una prueba, sus preguntas y las posibles respuestas y se persisten en base de datos. Se añade una nueva prueba asociada a la clase.
  - Selección de la prueba: el profesor accede a la prueba que acaba de crear.
    - Resultado: se navega a la vista de esa prueba. Se muestra el cuestionario. La prueba está inactiva, no hay participantes y las gráficas están vacías.
- Errores:
  - Fichero erróneo: el formato de fichero no es el correcto o presenta errores.
    - Resultado: se muestra una pantalla de error.

### 3.4.3. Caso de uso: participación en una prueba

- Resultado: se han registrado los resultados de participación de los alumnos y generado las gráficas de las preguntas. Se ha creado el ranking de la prueba y actualizado el global.
- Pasos a seguir:
  - Activación de la prueba: el profesor habilita la prueba.
    - Resultado: se registra la fecha de inicio. La prueba está disponible y es visible para los alumnos.
  - Resolución de la prueba: el alumno navega a la vista de pruebas y selecciona una. Para cada pregunta escoge una respuesta.
    - Resultado: se muestran las preguntas y opciones de respuesta junto a la caja desplegable para responder.
  - Corrección de la prueba: el alumno guarda sus respuestas.
    - Resultado: el corrector genera el resultado y lo persiste en base de datos. Se actualizan los parámetros individuales y del equipo junto a los correspondientes logros. El resultado no es visible aún.
  - Registro de la participación: el profesor revisa el progreso de la clase desde la vista de la prueba.
    - Resultado: aparecen registros con la participación en la tabla. Las gráficas se han actualizado en función de las respuestas dadas.
  - Finalización de la prueba: el profesor termina la prueba.
    - Resultado: se registra la fecha de fin. La prueba se marca como finalizada. Se genera el ranking de la prueba y se actualizan los logros/recompensas por posición en la clasificación. Los estudiantes ya pueden visualizar sus resultados desde la vista de juego.
- Errores:
  - Prueba finalizada sin participación: el profesor termina la prueba sin que ningún estudiante la completase.
    - Resultado: al no haber resultados no existe ranking de alumnos para esa prueba.



## Conclusiones y Trabajo Futuro

Language	files	blank	comment	code
Java	36	737	824	3098
HTML	14	288	6	1171
SQL	1	87	41	238
CSS	2	12	14	187
JSON	4	0	0	167
Maven	1	5	6	95
Bourne Shell	1	15	7	59
Javascript	1	6	26	25
SUM:	60	1150	924	5040

Figura 4.1: Lenguajes utilizados en el proyecto y sus líneas de código

El balance del desarrollo es positivo. El resultado final es una plataforma para gamificación de pruebas en el aula operativa, testeada y lista para su uso en el ámbito educativo (ver sección 3.4). También podría permitir la implantación de *blended learning* ya que el acceso y utilización de la plataforma es posible tanto desde el aula como desde el hogar. Se espera que la utilización de este proyecto no solo pueda incrementar la motivación de los alumnos sino que también pueda servir como herramienta de apoyo a los docentes para la evaluación del progresos de sus clases.

El desarrollo se ha llevado a cabo intentando seguir siempre las buenas prácticas pertinentes. Teniendo en cuenta las dependencias existentes en el modelo de datos se realizó una planificación priorizando las funcionalidades principales. La construcción del proyecto siguiendo un proceso incremental ha permitido un crecimiento fluido del sistema (ver sección 1.4), siendo mínimos los bloqueos debidos a dificultades relacionadas con la implementación.

En cuanto a la programación, se ha intentado conseguir la máxima modularidad para que el código sea sencillo de comprender (Fig. 3.9). La implementación de los controladores ha sido descompuesta en múltiples clases según la funcionalidad y el tipo de usuario al que atienden. Para facilitar la mantenibilidad se ha documentado

exhaustivamente cada controlador- Además, todos los parámetros utilizados por las herramientas para generación y carga de ficheros han sido recogidos en clases de constantes.

A través de la interfaz gráfica, construida a partir de HTML, CSS y Javascript, han conseguido reproducirse exitosamente las interacciones definidas en el desarrollo: la carga de ficheros, la descarga de credenciales en PDF, la formación de equipos y la resolución de pruebas. A pesar de no haberse empleado una librería para creación de interfaces responsive la utilización de la plataforma desde dispositivos portátiles sigue siendo posible y el menor tamaño de la pantalla no supone un impedimento.

Las tecnologías empleadas en la plantilla Java de la asignatura de “Ingeniería web”, utilizada como base para la construcción del proyecto, han acelerado en gran medida el desarrollo. Partiendo de la configuración ya definida ha sido posible centrarse en la implementación del sistema. Thymeleaf, el motor de plantillas de HTML, ha resultado fundamental en la construcción de la interfaz al permitir la fácil manipulación de la información del sistema. Del mismo modo, han sido de gran utilidad las opciones de seguridad de Spring Security para gestionar las sesiones de los usuarios. La persistencia de clases Java a través de Hibernate y JPA ha permitido de manera sencilla la generación del modelo de datos y la ejecución de las sentencias de consulta a la base de datos desde el propio código. Además de estas tecnologías, se han incorporado a través de Maven, y sin ninguna complicación, las librerías de ZXing, para generación de códigos QR, e ITextPDF, para la creación de documentos PDF. Trabajar desde Eclipse IDE ha facilitado la corrección de los errores que han aparecido durante la implementación y la navegación entre diferentes ficheros.

El control de versiones de todos los recursos generados durante el desarrollo del proyecto se ha realizado a través de Github, donde se han creado dos ramas adicionales. “desarrollo” sirve como copia de seguridad de la rama “master” y “modelo” es la rama utilizada para el trabajo diario. Mergeando a las ramas principales al tener una versión completa se ha simplificado el comprobar la evolución del proyecto respecto a la plantilla. Github también ha sido útil para anotar las tareas a realizar así como aquellas en proceso o ya completas. Para facilitar la revisión y corrección de la memoria se ha hecho uso de Overleaf, un editor online de Latex, lo que ha permitido la coordinación sencilla y eficaz entre alumno y tutor.

**RPClass** es código abierto, distribuido bajo la licencia “Apache 2.0”. Esta es una licencia que permite la modificación y distribución del software, debiendo aplicarse la misma licencia al producto resultante. Mediante esta licencia y la utilización de tecnologías actuales y altamente demandadas se espera favorecer la participación de la comunidad, la continuidad del proyecto más allá de su evaluación y el desarrollo de nuevas tecnologías similares. Para una explicación más profunda de la instalación y uso del sistema consultar los apéndices (Ap. A, Ap. B) o la **documentación del proyecto en Github**. El sistema puede incluso ser de interés como objeto de análisis y queda a disposición de la asignatura optativa del Máster “E-Learning”. impartida por Manuel Freire, director de este trabajo.

## **Estudio sobre la utilización del sistema**

A pesar de la valoración positiva de la plataforma el desarrollo no ha sido del todo satisfactorio y quedaría pendiente la realización de pruebas con usuarios reales. Aunque el funcionamiento del sistema sea el esperado faltaría comprobar el impacto de su aplicación en un escenario real para verificar su utilidad como herramienta de apoyo y medio para motivar a los alumnos. Debido a las complicaciones de la situación actual de pandemia mundial por Coronavirus (cuyo pico inicial en Madrid ocurrió a finales de marzo de 2020), y al estado de alarma declarado en mitad del desarrollo no han podido llevarse a cabo dichas pruebas y la experiencia quedaría como trabajo futuro a realizar.

Inicialmente se planteó el llevar a cabo un experimento piloto 1.4) con la colaboración de alumnos de educación secundaria, idealmente separando una misma clase de forma aleatoria en dos grupos. Puesto que la implementación actual de la plataforma se ha diseñado entorno a pruebas con preguntas tipo test, el sistema es especialmente adecuado para materias teóricas tales como la biología o la enseñanza de idiomas. A ambos grupos se les plantearían las mismas pruebas. Los integrantes de uno las resolverían manualmente en formato impreso, mientras que en el otro se utilizaría la versión gamificada.

Antes de llevar a cabo el experimento se contactaría a los profesores para orientarles e informarles sobre el uso y funcionamiento de la plataforma y se les pediría que preparasen una prueba de su asignatura. La experiencia tendría una duración de una semana a lo largo de la cual se les plantearían diferentes pruebas a los alumnos. En el primer día se presentaría la plataforma a los estudiantes y se dedicaría una sesión a explicar el funcionamiento de la misma, organizar la actividad y preparar el escenario. Para asegurar el esfuerzo y la dedicación del alumno un porcentaje de la nota se correspondería con la participación en las pruebas. A mediados de la semana se invertirían los roles de los grupos, pasando cada uno a la otra modalidad de resolución.

Al final de la semana se realizaría a los estudiantes una encuesta de satisfacción sobre las pruebas como método de evaluación y su experiencia, tanto respecto al método gamificado como a las pruebas resueltas a mano. Esta encuesta valoraría la opinión sobre la dificultad de las pruebas y su utilidad para mantener unos hábitos de estudio continuo así como sobre la usabilidad y aspecto de la plataforma. Al final de la experiencia se compararía la motivación y satisfacción de los estudiantes según el método de evaluación así como las calificaciones obtenidas por cada grupo y las tendencias que han mostrado.

Del mismo modo se realizaría una encuesta a los profesores implicados para conocer su opinión sobre la plataforma y las dificultades que hubiesen podido encontrar en la organización de las actividades o la generación de los contenidos. Siendo conscientes de las limitaciones temporales que pueden producirse en la docencia, es fundamental minimizar el esfuerzo necesario por parte del profesorado para hacer uso del sistema.

En base a la información recopilada y a los resultados de las encuestas realizadas a alumnos y profesores se procederían a realizar las modificaciones y ajustes pertinentes para conseguir una experiencia más satisfactoria. Para continuar contrastando las capacidades y el alcance del sistema, se podrían realizar experiencias adicionales extendiendo el estudio a otros niveles educativos como Bachiller y enseñanza universitaria.

## Trabajo futuro

Como posibles mejoras, muchas de las cuales se barajaron durante el diseño de la aplicación, vale la pena mencionar:

**Apartado gráfico** La principal modificación, y una de las que más podría beneficiar a la aplicación, sería una mejor integración en dispositivos portátiles. Debido a las dificultades que entraña un buen diseño *responsive* y a la falta de experiencia previa la implementación de la interfaz gráfica precisaría de múltiples ajustes para una experiencia más satisfactoria. Otra alternativa podría ser la creación de una app móvil con un diseño adaptado a la plataforma, lo cual daría la oportunidad de integrar nuevas formas de interacción y funcionalidades.

**Organización de equipos** Respecto a la formación de los grupos, dicha tarea podría automatizarse o convertirse en algo más entretenido. Una de las opciones iniciales que se valoró fue el equilibrado de los equipos teniendo en cuenta las puntuaciones de los alumnos. Esta se descartó debido a la necesidad de un uso prolongado de la aplicación para la diferenciación de las capacidades de cada estudiante. Otra propuesta fue relegar la tarea sobre los alumnos, nombrando un capitán por equipo y siendo estos quienes vayan escogiendo a sus compañeros.

**Variedad y complejidad de las pruebas** Las pruebas también tienen margen de mejora, por ejemplo, mediante la incorporación de material audiovisual, en lugar de limitarse a preguntas en formato textual, o la creación de ejercicios más interactivos. También podrían introducirse ajustes en el sistema de evaluación de las mismas. Siempre permitiendo que los docentes tengan el mayor control posible sobre las pruebas y minimizando la complejidad de los ficheros y el esfuerzo necesario para la generación de las mismas. Una implementación más costosa aunque puede que más cómoda al profesorado sería la integración de un editor de clases y pruebas dentro de la propia plataforma.

**Análisis estadístico** En base a todos los datos recolectados durante la resolución de las pruebas sería interesante la generación de un análisis estadístico que ilustre el progreso individual de los estudiantes y permita dedicarle una atención personalizada ante al detectar resultados inusuales. A través de esta información podría incluso generarse distintos perfiles de estudiante que en conjunto con los resultados de las pruebas podrían ayudarnos a inferir la dificultad de las preguntas y detectar tendencias.

**Alianzas** Para fomentar aún más la colaboración y competitividad entre los alumnos se planteó la participación conjunta de los equipos para equilibrar las clasificaciones. Mediante alianzas los equipos con mejores puntuaciones podrían apoyar en las pruebas a los más rezagados y ayudar a su escalada en el ranking, consiguiendo una mayor movilidad en las clasificaciones y una mayor dificultad para mantener la posición.

**Sistema de comunicación** Sería interesante integrar un foro donde los estudiantes pudiesen consultar sus dudas y chats grupales para el intercambio de mensajes dentro del equipo. También sería práctico un sistema de notificaciones alertando del inicio/fin de las pruebas y del progreso del alumno al desbloquear logros o conseguir buenos resultados. Una posibles solución sería, por ejemplo, la integración de un widget de una aplicación externa como podría ser Telegram o Discord.

**Extensión del dominio** Sería deseable ampliar el ámbito de la aplicación, permitiendo compartir una misma cuenta de estudiante para varias asignaturas. Esto aumentaría el alcance de la gamificación propuesta, dando una mayor intensidad a la competición y permitiendo a los alumnos compensar las carencias de una asignatura con resultados de otra que dominen más. También sería interesante poder evaluar al profesorado a través de la plataforma, concediendo logros y recompensando los buenos resultados conseguidos por sus alumnos. Debido a la complejidad del diseño de estas funcionalidades y al esfuerzo necesario para una implantación exitosa esta propuesta queda como línea de trabajo a largo plazo como orientación del proyecto. Una forma de abordar la implementación de esta mejora sería la adaptación de la plataforma para cumplir con los estándares LTI (IMS Global, 2020). De este forma sería posible la integración con LMS (por ejemplo Moodle), facilitándose la implantación en la infraestructura tecnológica de las instituciones educativas.



## Introduction

Technology has always been used as a supporting tool for education, providing new perspectives and facilitating access to those with difficulties (Raja y Nagasubramani, 2018). Educational technology is a reality present in the majority of our classrooms and its implantation is increasing by the day. For example, it is now easier than ever to learn from home thanks to the great quantity of available digital resources, with easy access to knowledge from a wide variety of platforms.

Taking advantage of the potential of technology, multiple alternatives supporting learning have appeared, from online courses to e-learning platforms. From my experience as a student I have been able to see how the atmosphere in the class changes while performing an activity that breaks routine. Such activities are usually related with the use of an electronic device or software. Interaction among classmates increases and they participate more frequently than usual, especially if the activity implies any kind of competitive factor. While studying my degree some of these experiences included use of *clickers*, Kahoot! contests, and code competitions with automatic grading. The approach proposed by this project is focused on gamification, the application of game mechanics to educational-professional fields in order to improve motivation and performance.

### 5.1. Motivation

Among the multiple applications of technology in education, gamification is generating considerable interest, by proposing a more stimulating way of teaching, in which game mechanics are used to improve the obtained results. Adapting game characteristics to education, a more practical, interactive and entertaining approach can be achieved (Gee, 2005).

A fundamental factor to take into account when teaching is to keep students motivated. New technologies have proven to, adequately used, turn learning into an attractive and accessible task and therefore a more motivating one (Francis, 2017).

The use of electronic devices as a complement to traditional teaching methods is getting more frequent day by day. Many studies have analyzed if the use of technology while learning carries better results and an increase of interest and engagement in students (Hainey et al., 2016).

The combination of face-to-face teaching, in which students receive lessons from a teacher in a classroom, with the use of electronic devices and educative online resources is known as *blended learning* or *b-learning* (Bartolomé, 2004). Taking the advantages of both modalities is possible to expand education beyond the classroom and facilitate the access of students to learning material. However, it is necessary to take into account the investment in time and resources to technologically train teachers in the organization and management of teaching models with *blended learning*.

**RPClass** takes that scenario as a base and proposes the gamification of classes through a web application. Using this platform, teachers will be able to set contests for their students. Students can then participate through any kind of device with Internet access while they compete for obtaining the best results. The project has been implemented as an experimental support tool for teaching, with the goal of stimulating student participation.

The application can be especially useful in subjects requiring of the memorization of great quantities of theoretical information. As these do not need complex problem-solving support, adapting the contest to the system is a more affordable task. The gamification of these purely theoretical subjects can provide a more dynamic and engaging perspective, avoiding monotony and loss of motivation in students. Through the organization of contest students can be encouraged to acquire constant study habits it can be checked if the class is understanding properly the content taught in classes.

One of the existing characteristics of **RPClass**, which is expected to stimulate the mutual support among students in the learning process, is the cooperative approach of the gamification. Students, organized in groups, will have to collaborate in teams to reach the highest position in the classification. By setting a common goal, we expect a collective effort where every student tries to progress and improve results along with their teammates. The final result will depend on the engagement of each member. Studies have proved positive effects of allowing to students to manage each other results in the fair evaluation of individual contribution within the group (Moccozet et al., 2013).

## 5.2. Objectives

The purpose of this project is the development of a web application for exercises gamification in the classroom. For that, the test used by teachers as an evaluation method will be presented with a different approach. The solving of these will be transferred to the web application and students completing them will receive a score according to their performance. Students will be organized in teams and will participate in a competition to win point and unlock achievements.



Each achievement acknowledge the fulfillment of goals, such as reaching a quantity of passed contest or questions answered correctly. For each contest, the results of the students will be collected and statistic graphics for each question will be generated according to the answers.

The system has been designed to be used in university and high-school classrooms with the required technology available for teacher and students. The most usual scenario is expected to be a teacher setting a contest from the PC and the students solving it from their laptops or portable devices. We also envision a scenario in which a teacher organizes a contest and makes it available for the students to solve at home.

Taking the previous approach for the application and the most usual scenarios into account, the following objectives must be accomplished:

- Easy interaction with the system: teachers must be able to easily create their own classes and contest, as well as organize the students into teams. Students must be able to easily solve the proposed contests.
- Solving contests: students must be able to participate from the platform in the contest created by the teachers. Availability of contests must be managed.
- Evaluating the results: to simplify the work of teachers, contest must be automatically corrected. Furthermore, following the progress of the classes must be facilitated.

### 5.3. Development model

The most suitable development model according to the nature and characteristics of the project is an iterative and incremental model (Fig. 1.1). This one allows an easy adaptation to potential changes as well as a short term obtainment of results with each iteration. A division of the project can be made according to the two main user roles, the logic related to teachers and the one related to students. Additionally, we can classify the implementations depending on the data model elements intervening.

Given this approach, the development will be structured into three iterations. The first one will include the design phase, the second will correspond to the implementation and during the third one the documentation will be elaborated. The second iteration will be composed by many cycles and at the end of each a prototype incorporating new tested functioning characteristics will be obtained. Limiting and completing functionalities before moving to the next cycle simplifies version control. This way, in case of an unexpected error it will be easier to identify and correct it achieving a smoother development and minimizing blocking risks.

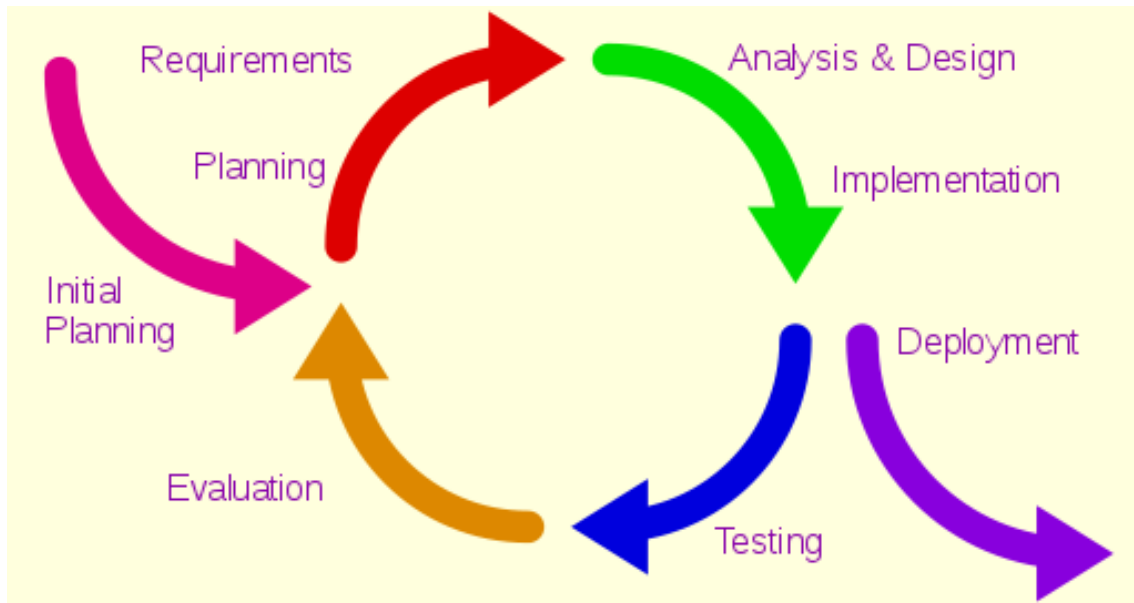


Figure 5.1: Flow of an iterative and incremental development model. Wikimedia Commons (2014)

## 5.4. Work plan

Given the lack of previous experience in this kind of projects and in anticipation of possible difficulties appearing during the development, most of the available time will be invested in the implementation phase. In order to minimize the risk, the design phase will extend until all the factors involving the development have been properly defined. Finally, the remaining time will be destined to the documentation of the project and the preparations for its distribution. Take that scenario into account, the planning estimates the following deadlines by phase and tasks to solve:

- Design phase:
  - Duration: 4 weeks. 04/02/20 a 02/03/20.
  - Cycle 1: Requirements specification: 1 week
    - Tasks:
      - ◊ Define objectives and requirements
      - ◊ Define architecture and system components
  - Cycle 2: Select technologies: 1 week
    - Tasks:
      - ◊ Analyze and select technologies
  - Cycle 3: Graphic design: 1 week
    - Tasks:
      - ◊ Design views
      - ◊ Design navigation and interaction

- ◊ Create *mockups*
- Cycle 4: Data model design: 1 week
  - Tasks:
    - ◊ Define data model classes
- Implementation phase:
  - Duration: 8 weeks. 03/03/20 to 27/04/20
  - Cycle 1: creation of classes: 2 weeks
    - Tasks:
      - ◊ Process file
      - ◊ Organize teams
  - Cycle 2: Contest creation: 2 weeks
    - Tasks:
      - ◊ Process file
      - ◊ Build contest to be solved
  - Cycle 3: Contest participation: 2 weeks
    - Tasks:
      - ◊ Gather results
      - ◊ Generate graphs
      - ◊ Generate rankings
  - Cycle 4: Include achievements: 1 week
    - Tasks:
      - ◊ Create achievements
      - ◊ Update achievements after contests
  - Cycle 5: Testing: 1 week
    - Tasks:
      - ◊ Check use cases
      - ◊ Correct interface
      - ◊ Experiment with real users
- Documentation phase: 28/04/20 a 08/06/20
  - Duration: 6 weeks.
  - Tasks:
    - Prepare the report
    - Document code
    - Use guides
    - Wiki in Github
- Documentation phase:
  - Duration: 6 weeks. 28/04/20 a 08/06/20

- Cycle 1: Final report: 3 weeks
  - Tasks:
    - ◊ Write the report
    - ◊ Compile bibliography
    - ◊ Diagram generation
- Cycle 2: Document code: 1 week
  - Tasks:
    - ◊ Comment source code
    - ◊ Merge final version in repository
- Cycle 3: Use guides: 1 week
  - Tasks:
    - ◊ Use guide
    - ◊ Installation guide
- Cycle 4: Wiki in Github: 1 week
  - Tasks:
    - ◊ Document project
    - ◊ Add use guides

The remaining weeks until the first submission (26/06/20) will remain unassigned in order to provide flexibility to the development and as a margin in anticipation of possible unexpected difficulties.

Throughout this work, I held regular meetings with the tutor to control the progress of the project. They were more frequent during the initial design phase and during the last documentation phase. The meetings were arranged every two weeks approximately and the frequency of these increased to weekly as the end of the development drew near.

## Conclusions and Future Work

Language	files	blank	comment	code
Java	36	737	824	3098
HTML	14	288	6	1171
SQL	1	87	41	238
CSS	2	12	14	187
JSON	4	0	0	167
Maven	1	5	6	95
Bourne Shell	1	15	7	59
Javascript	1	6	26	25
SUM:	60	1150	924	5040

Figure 6.1: programming languages used in the project and lines of code

The balance of the project is positive. The final result is an working and tested platform for class gamification ready for its use in an educational field (see section 3.4). It could also allow implementation of *blended learning*, as the platform can be accessed and used both from a classroom or from home. Use of this project is expected to increase motivation among students as well as to serve as a support tool for teachers in the evaluation of their progress.

We have followed best practices during development. Taking into account the dependencies within the data model, a plan was made prioritizing the main functionalities. The project was built according to an incremental process, which has allowed the smooth progression of the system (see section 5.4), keeping the implementation-related impasses to the minimum.

Regarding the programming, we have tried to achieve high code modularity, while making it as simple and understandable as possible (Fig. 3.9). The implementation of controllers has been decomposed into multiple classes according to the functionality and the type of user they serve. To facilitate maintainability, every controller has been exhaustively documented. Moreover, all the parameters used in the generation and file formats has been collected into constants classes.

Through the graphic interface, built with HTML, CSS and Javascript, all the interactions defined in the design have been successfully replicated: loading files, downloading the access credentials in PDF format, organizing the teams and solving the contests. Despite not using a library to build responsive interfaces, the use of the platform from portable devices is still possible and the smaller screens are not an issue.

The technologies used in the Java template from the subject “Web Engineering”, used as the basis to build the project, has greatly sped the development. Starting from the initial configuration it has been possible to focus on the implementation of the system. Thymeleaf, the HTML template engine, has been essential while building the interface as it allows to easily manipulate the information in the system. Similarly, the security options provided by Spring Security have proved themselves to be really useful in the management of the sessions of the users. Persisting Java classes with Hibernate and JPA has allowed to generate the data model and execute queries to the database straight from the code. Apart from these technologies, more libraries have been included through Maven with no complications. ZXing, which is used to generate QR codes, and iTextPDF, which is used in the creation of PDF files. Working from Eclipse IDE has simplified the correction of the errors appearing during the implementation and the navigation between different resource files.

Version control of the resources generated in the development of the project has been managed through Github, where two additional branches have been created. “desarrollo” serves as backup for “master” and “modelo” is used to commit the daily work. Merging the main branches with a complete version simplifies the task of checking the evolution of the project respect to the template. Github has also been useful to keep track of the pending tasks as well as the completed ones and the ones in progress. To ease the revision and correction of the report Overleaf, an online Latex editor, has been used. That has allowed a simple and effective coordination between the student and the director of the project.

**RPClass** is open source distributed under the license “Apache 2.0”. This is a license that allows modification and distribution of the software, being mandatory to apply the same license to the resulting product. By the use of this license and modern highly demanded technologies it is expected to stimulate the engagement of the community, the continuity of the project beyond its evaluation and the development of similar technologies. For a further explanation of the installation and use of the system see the appendixes (Ap. A, Ap. B) or the **documentation of the project in Github**. The system may even be an interesting subject of analysis in the optional subject of the Master “E-Learning”, taught by Manuel Freire, director of this project.

## Study of the use of the system

Despite the positive assessment of the platform, testing the system with real users remains as a pending task. Although it works as expected in tests, the impact of using the application in a real scenario, and verify its use as a support tool and means to motivate the students, is still to be proven. Due to the complications of the current

global situation caused by the COVID-19 pandemic (which reached its initial peak in Spain by the end of March 2020), and to the state of alarm declared in the middle of the development it has not been possible to carry out such studies, and the experiment remains as future work.

Initially, we intended to carry out a pilot experiment 5.4) with the collaboration of high-school students, ideally separating them randomly into two groups. Given that the current implementation of the platform has been design around test type questions, the system is especially fitting for theoretical subjects such as Biology or Languages. Both groups would be presented with the same tests. The members of one would solve them manually in a printed format, while the others would try the gamified version.

Before carrying the experiment teachers would be contacted to guide and inform them about the use and operation of the platform and they would be asked to prepare a contest for their subject. The experiment would last a week, during which several contests would be presented to the students. On the first day the platform would be introduced to the students and a session would be destined to explain how it works, organize an activity and set the scenario. To grant the effort and dedication of the student a percentage of the grade would correspond with the participation in the contests. In the middle of the week roles would be swapped between groups, passing each of them to the other solving modality.

At the end of the week students would be queried about their satisfaction with the contest as an evaluation method and their experience with the gamified method as well as with the manually solved contests. This poll would evaluate the opinions about the difficulty of the contests and the utility of these to keep constant study habits as well as about the usability and appearance of the platform. At the end of the experiment the motivation and satisfaction of the students would be compared between the evaluation methods. Same would be done with the grades obtained by each group and the tendencies they show.

In the same way, a poll would be presented to the involved teachers to know their opinion about the platform and the difficulties they could have found in the organization of the activities or in the creation of the resources. Being aware of the issue that limited time is in teaching, it is fundamental to minimize the necessary effort for teachers to make use of the system.

According to the collected information and to the results of the polls made to students and teachers the proper modifications and adjustments would be made to achieve a more satisfactory experience. To continue contrasting the capacities and reach of the system, additional experiments could be made expanding the study to other educational levels such bachelor and university teaching.

## **Future work**

As possible improvements for the system, many of which were considered during the application's design, the following are especially worth mentioning:

**Graphic aspects** The main modification, one of those which could benefit more to the application, would be a better integration with portable devices. Due to the difficulties regarding *responsive* designs and the lack of previous experience the graphic interface would need of multiple adjustments for a more satisfactory experience. Another alternative would be the development of a mobile app with a design adapted to the platform, which could give the opportunity to integrate new ways of interaction and functionalities.

**Team organization** In regard to the group formation, this task could be automatized or transformed into something more entertaining. One of the initial options was to build balanced teams according to the individual scores of their students. This one was discarded due to the necessity of using of the application for a extended period of time properly differ each student capacities. Another alternative was to delegate this task to the students, choosing a captain per team and being those the ones to pick their teammates.

**Variety and complexity of the contest** There is room for improvement of the contests, for example, by including audiovisual material instead of limiting to textual questions or by creating more interactive exercises. Adjustments in the evaluation system of the contest could also be introduced. Always allowing teachers to be in control of the contests and minimizing the complexity of the resource files and the necessary effort for the creation of these. Although costly, a more convenient alternative for the teachers would be the integration of a class and contest editor within the platform.

**Statistic analysis** Based on the data collected during contests, it would be interesting to generate a statistical analysis showing the individual progress of the students to give each a personalize treatment in case of detecting unusual results. Through this information it could even be possible to generate different student profiles that along with the obtained results could hep to measure the difficulty of the questions and find tendencies in the answers.

**Alliances** To stimulate even more the collaboration and competitiveness among students the jointly participation of teams to level the classifications was planned. Through alliances the teams with better scores could support those in the lowest positions during the contests and help to climb the ranking. This way, a greater mobility in the classifications and a greater difficulty to maintain the position is achieved.

**Communication system** It would be interesting to integrate a forum where students could solve their doubts and group chats for teams to exchange messages. It could also be useful to implement a notification system alerting about the start/end of contests and about the progress of the students when unlocking achievements or obtaining good results. A possible solution for this task, for example, could be the integration of a widget from an external application such as Telegram or Discord.

**Domain expansion** It would be desirable to expand the range of the application, allowing a single account to be shared through several subjects. These would increase the reach of the gamification, intensifying the competition and allowing



students to compensate the scores in a subjects with the results of other in which they are more skilled. It would also be interesting for teachers to be evaluated through the application by giving them achievements and rewarding the good results obtained by their students. Due to the complexity involving the design of these functionalities and to the necessary effort for a successful implantation this approach remains as a long term goal to guide the project towards. A good alternative to tackle the implementation of this improvement would be the adaptation of the platform to comply with the LTI (IMS Global, 2020) standards. That way it would be possible to integrate it with LMS (for example, Moodle), facilitating the implantation within the technological infrastructure of educational institutions.



# Manual de instalación

**Repositorio del proyecto:** <https://github.com/Aitorcay/RPClass>

La configuración del sistema viene definida en tres ficheros:

- `application.properties`: configuración general del sistema
- `application-default.properties`: crea una base de datos para cada ejecución
- `application-externaldb.properties`: utiliza una base de datos externa
- `server.properties`: configuración del servidor HSQLDB

## A.1. Configuración del directorio de ficheros

El directorio en el que se almacenarán los archivos utilizados y generados por la aplicación se define en el fichero `application.properties`.

---

```
es.ucm.fdi.base-path: /tmp/iw
```

---

Listado A.1: Definición del directorio de la aplicación

En dicha dirección se crearán las carpetas con las imágenes de perfil de los usuarios y equipos y los ficheros con los QR creados por los profesores. Aquí se incluirán otros recursos externos como podrían ser las plantillas a disposición de los docentes para creación de contenido.

## A.2. Configuración de la base de datos

La definición de la base de datos se realiza en el fichero de `application.properties`.

---

```
spring.profiles.active: default
```

---

Listado A.2: Definición de la base de datos

La propiedad puede tomar dos valores

- **default:** se utilizará el fichero `application-default.properties` y se creará una nueva base de datos.

---

```
spring.datasource.url: jdbc:hsqldb:mem:iw;create=true
```

---

Listado A.3: Creación de una base de datos temporal en memoria

- **externaldb:** se utilizará el fichero `application-externaldb.properties` y se utilizará una base de datos externa, la cual se espera que sea una HSQLDB y esté disponible en `localhost`.

---

```
spring.jpa.hibernate.ddl-auto: create-drop  
spring.datasource.url: jdbc:hsqldb:hsqldb://localhost/iwdb
```

---

Listado A.4: Utilización de una base de datos externa

Las propiedades por defecto para el servidor de la HSQLDB están definidas en el fichero `server.properties`:

---

```
server.database.0=file:/tmp/iwdb  
server.dbname.0=iwdb  
server.remote_open=true
```

---

Listado A.5: Definición del servidor de la HSQLDB

En caso de que se desee incluir en la base de datos información por defecto puede utilizarse el fichero `import.sql`. Este cargará su contenido al iniciarse la aplicación si es correcto y respeta las restricciones definidas por el modelo de datos.

### A.3. Ejecución de la aplicación desde Eclipse

Requisitos previos:

- Maven
- JDK 8

Para comenzar a utilizar la plataforma bastará con seleccionar y ejecutar la clase `IwApplication.java` (List. A.6).

Una vez lanzada, la aplicación se estará ejecutando en `localhost:8080` (Fig. A.2) y podrá accederse cualquier navegador.

Run > Run as > Java application

Listado A.6: Ejecución de IwApplication.java

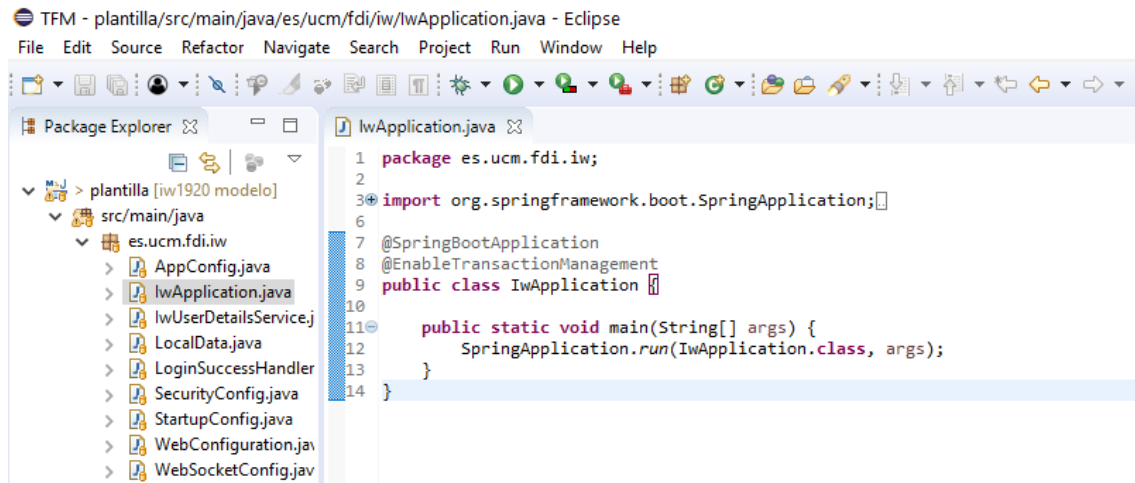


Figura A.1: Localización de IwApplication.java en el proyecto

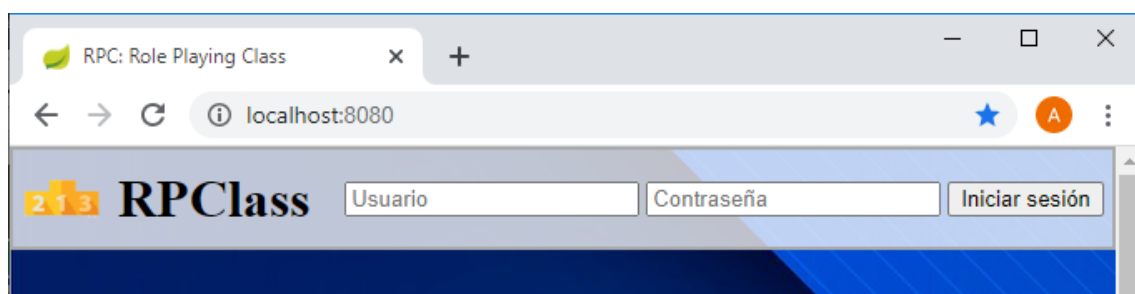


Figura A.2: Acceso a la plataforma desde el navegador



## Manual de uso

### B.1. Creación de una clase

Para la creación de una nueva clase será necesario rellenar el correspondiente fichero JSON. Se proporcionan una plantilla y un ejemplo junto con el proyecto.

```
{
  "nombreClase": "escribe aqui el nombre de la clase",
  "alumnos": [
    {
      "nombre": "escribe aqui el nombre del estudiante",
      "apellidos": "escribe aqui los apellidos del estudiante"
    }
  ]
}
```

Listado B.1: Plantilla de fichero para cargar una clase

Simplemente será necesario rellenar el nombre de la clase y los nombre y apellidos de los alumnos. Una clase podrá incluir tantos alumnos como se desee, separados entre sí por comas.

```
{
  "nombreClase": "Clase de ejemplo",
  "alumnos": [
    {
      "nombre": "Gloria",
      "apellidos": "Cebrian Pereiro"
    },
    {
      "nombre": "Agustin",
      "apellidos": "Grande Borrego"
    }
  ]
}
```

```
},  
{  
  "nombre": "Maria",  
  "apellidos": "Burgos Acunia"  
}  
]  
}
```

Listado B.2: Fichero con una clase de ejemplo

El siguiente paso, una vez rellenado el fichero, será cargarlo desde la vista de clases. Solamente las cuentas de los docentes tienen acceso a esta funcionalidad.

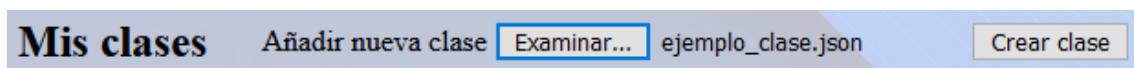


Figura B.1: Fichero de clase seleccionado

Tras procesarse el fichero se mostrará la siguiente vista. La caja de la izquierda contiene un panel por cada estudiante especificado en el fichero. La caja de la derecha permite crear un los contenedores de los equipos.

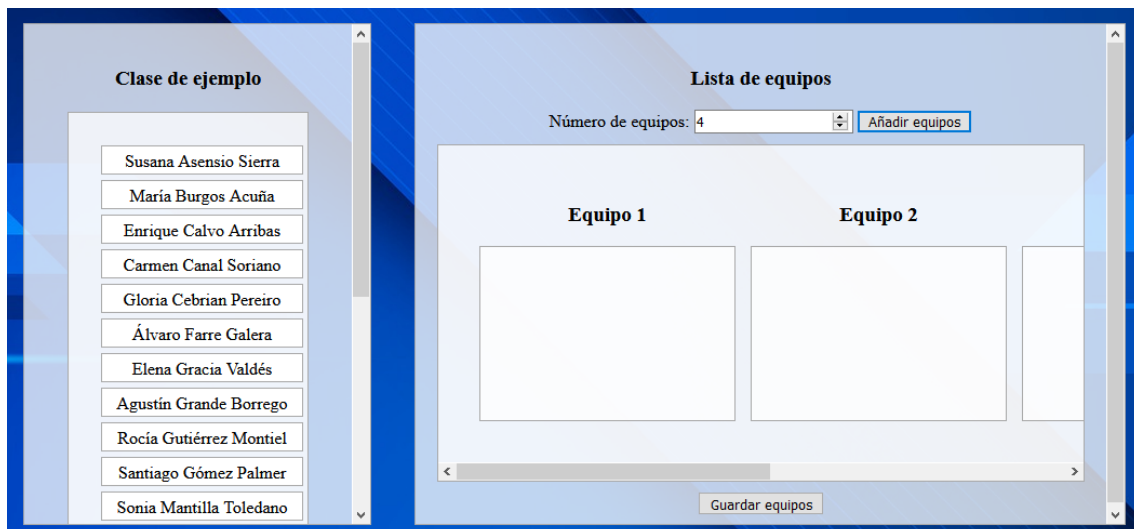


Figura B.2: Vista de clases tras la carga de fichero

Después de indicar el número de equipos simplemente habrá que pinchar en los paneles de los alumnos y arrastrarlos hasta el correspondiente grupo. Una vez organizados todos los alumnos guardar los cambios para terminar el proceso.

Una vez completados todos los pasos se añadirá la clase a la lista de creadas por el docente. La vista de esa clase incluirá la lista de alumnos, la lista de equipos organizados y un panel desde el que añadir las pruebas asociadas a esa clase.



Figura B.3: Organización de los alumnos en equipos

Figura B.4: Vista de clases tras completar el proceso de creación

## B.2. Creación de una prueba

Para la creación de una nueva prueba será necesario rellenar el correspondiente fichero JSON. Se proporcionan una plantilla y un ejemplo junto con el proyecto.

```
{
  "nombreConcurso": "escribe aqui el nombre de la prueba",
  "preguntas": [
    {
      "enunciado": "escribe aqui el enunciado de la pregunta",
      "respuestas": [
        "escribe aqui una opcion-->escribe aqu el porcentaje de
          puntuacin (un valor entre -1 y 1)",
        "escribe aqui una opcion-->escribe aqu el porcentaje de
          puntuacin (un valor entre -1 y 1)",
        "escribe aqui una opcion-->escribe aqu el porcentaje de
```

```

    puntuacin (un valor entre -1 y 1)",
    "escribe aqui una opcion-->escribe aqu el porcentaje de
    puntuacin (un valor entre -1 y 1)"
  ]
}
]
}

```

Listado B.3: Plantilla de fichero para cargar una prueba

Simplemente será necesario rellenar el nombre de la prueba y las preguntas. Cada pregunta consta de un enunciado y un conjunto de opciones con las que responder. Las opciones incluyen el texto y su puntuación separados por -->(sin espacios). La puntuación es un valor entre -1 y 1 que representa el porcentaje de acierto.

- 1: respuesta correcta. Suma el 100 % de puntuación de la pregunta.
- 0: respuesta incorrecta. 0 % de puntuación de puntuación.
- 0 a 1: respuesta parcialmente correcta. Suma un % de puntuación de la pregunta.
- -1 a 0: respuesta incorrecta con penalización. Resta un % de la puntuación de la pregunta.

```

{
  "nombreConcurso": "Concurso de prueba",
  "preguntas": [
    {
      "enunciado": "Cual de las siguientes afirmaciones sobre las
        arquitecturas paralelas existentes es falsa",
      "respuestas": [
        "La arquitectura ms potente del mundo ofrece un rendimiento
          del orden de decenas de Petaflops-->0",
        "Para las aplicaciones de alto rendimiento acopladas se
          requiere una red de interconexion de baja latencia-->0",
        "Segun el Top500 la arquitectura ms comun es la memoria
          compartida-->1",
        "El uso de una infraestructura Cloud no implica realizar
          cambios en las aplicaciones-->0"
      ]
    },
    {
      "enunciado": "Cual de las siguientes afirmaciones sobre tipos de
        aplicaciones es falsa?",
      "respuestas": [
        "Las aplicaciones de alta productividad no requieren el uso
          de un modelo de programacion paralela-->0",

```

```
"El modelo de alta productividad es compatible con el de alto
rendimiento ya que cada tarea puede requerir el uso
simultaneo de varios procesadores-->0",
"Los flujos de trabajo (workflows) son un ejemplo de
aplicacion de alta productividad-->0",
"Las aplicaciones de alta productividad son solo aquellas que
se ejecutan sobre arquitecturas de ejecucin
oportunistas-->1"
]
}
]
```

Listado B.4: Fichero con una prueba de ejemplo

El siguiente paso, una vez rellenado el fichero, será cargarlo desde el panel de pruebas de la correspondiente clase. Las pruebas van asociadas a clases concretas.

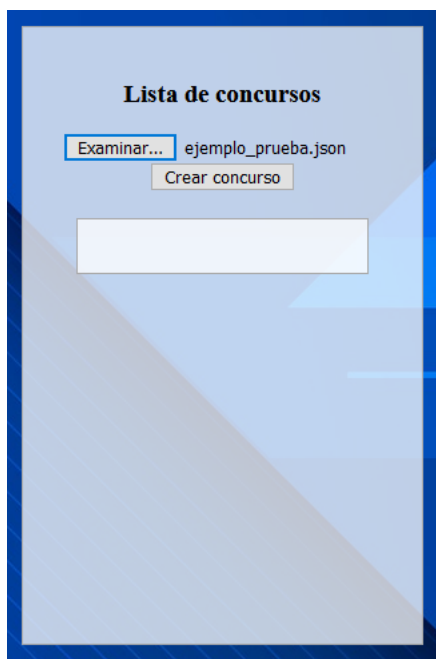


Figura B.5: Fichero de prueba seleccionado

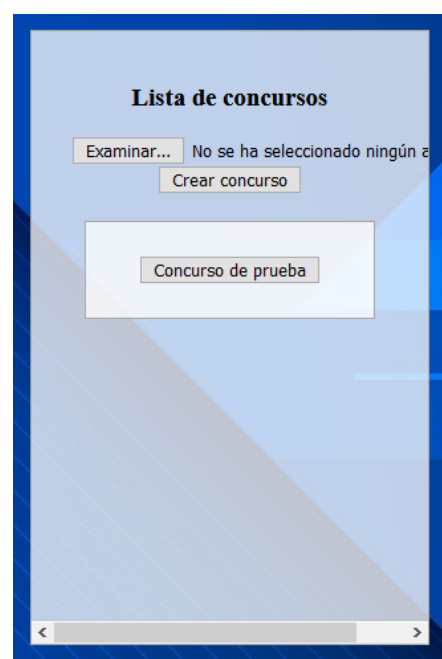


Figura B.6: Prueba cargada y añadida a la correspondiente clase

Tras procesarse el fichero se mostrará la prueba en el contenedor. Al hacer click sobre ella navegaremos hasta la vista de pruebas y se mostrará la información completa de la misma.

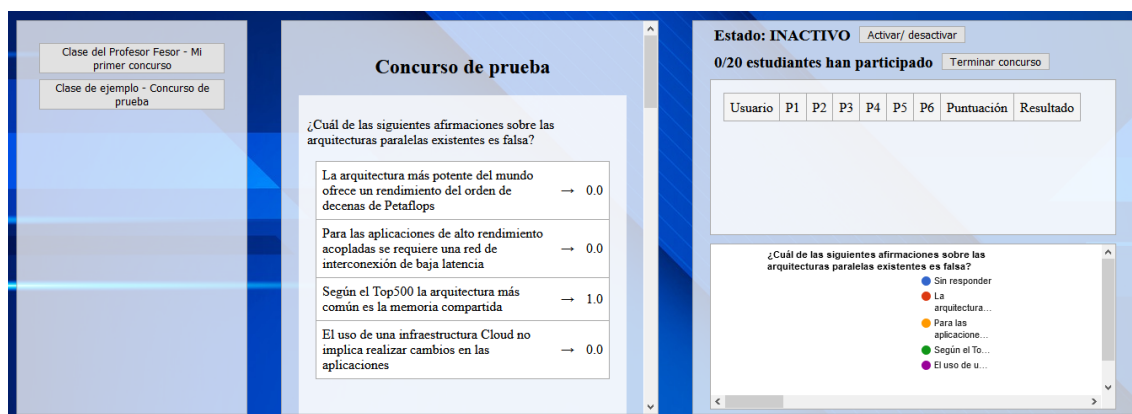


Figura B.7: Vista de una prueba recién creada

### B.3. Gestión de las pruebas

Una vez creada una prueba es necesario habilitarla para que los estudiantes tengan acceso y puedan participar.

A medida que los alumnos completan la prueba se irán recogiendo sus resultados en una tabla. En función de estos se generarán las gráficas asociadas a cada pregunta que muestran las estadísticas de las opciones escogidas para cada una.



Figura B.8: Resultados de una prueba y estadísticas de las preguntas

El docente podrá finalizar en cualquier momento la prueba. Cuando esto suceda se procederá a la generación de las clasificaciones de la prueba correspondiente y se

otorgarán las recompensas a alumnos y equipos por posición en el ranking.



Figura B.9: Rankings de estudiantes y equipos en una prueba



# Bibliografía

- Alsawaier, R. The effect of gamification on motivation and engagement. *International Journal of Information and Learning Technology*, vol. 35, páginas 00–00, 2017. Available at [https://www.researchgate.net/publication/321063416\\_The\\_Effect\\_of\\_Gamification\\_on\\_Motivation\\_and\\_Engagement](https://www.researchgate.net/publication/321063416_The_Effect_of_Gamification_on_Motivation_and_Engagement). Last access: 23-06-20.
- Ang, C. S. y Rao, R. Theories of learning: A computer game perspective. páginas 239 – 245. 2004. ISBN 0-7695-2031-6. Available at [https://www.researchgate.net/publication/4049495\\_Theories\\_of\\_learning\\_A\\_computer\\_game\\_perspective](https://www.researchgate.net/publication/4049495_Theories_of_learning_A_computer_game_perspective). Last access: 23-06-20.
- Angular. Angular. 2020. Available at <https://angular.io/>. Last access: 23-06-20.
- Barata, G., Gama, S., Jorge, J. y Gonçalves, D. Engaging engineering students with gamification. En , páginas 24–31. 2013. Available at [https://www.researchgate.net/publication/259821679\\_Engaging\\_Engineering\\_Students\\_with\\_Gamification](https://www.researchgate.net/publication/259821679_Engaging_Engineering_Students_with_Gamification). Last access: 23-06-20.
- Bartolomé, A. Blended learning. conceptos básicos. *Pixel-Bit*, 2004. Available at [https://www.researchgate.net/publication/277262026\\_Blended\\_learning\\_Conceptos\\_basicos](https://www.researchgate.net/publication/277262026_Blended_learning_Conceptos_basicos). Last access: 23-06-20.
- Bekele, T. Motivation and satisfaction in internet-supported learning environments: A review. *Educational Technology & Society*, vol. 13, páginas 116–127, 2010. Available at [https://www.researchgate.net/publication/220374959\\_Motivation\\_and\\_Satisfaction\\_in\\_Internet-Supported\\_Learning\\_Environments\\_A\\_Review](https://www.researchgate.net/publication/220374959_Motivation_and_Satisfaction_in_Internet-Supported_Learning_Environments_A_Review). Last access: 23-06-20.
- Brinkman, D. Just Press Play - Microsoft Research. 2011. Available at <https://www.microsoft.com/en-us/research/project/just-press-play/>. Last access: 23-06-20.
- Conaway, R. y Garay, M. Gamification and service marketing. *SpringerPlus*, vol. 3, página 653, 2014. Available at [https://www.researchgate.net/publication/268233579\\_Gamification\\_and\\_service\\_marketing](https://www.researchgate.net/publication/268233579_Gamification_and_service_marketing). Last access: 23-06-20.

- Cujba, S. LMS Comparison: What LMS suits your needs best? 2020. Available at <https://raccoongang.com/blog/lms-comparison-what-lms-suits-your-needs-best/>. Last access: 23-06-20.
- Daityari, S. Angular vs React vs Vue: Which Framework to Choose in 2020. 2019. Available at <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>. Last access: 23-06-20.
- DevTeam. JDBC vs. Hibernate: Which to Choose? 2020. Available at <https://www.devteam.space/blog/hibernate-vs-jdbc-which-to-choose/>. Last access: 23-06-20.
- Elizabeth Lawley. Just press play badge. 2012. Available at <https://clalliance.org/wp-content/uploads/2012/10/JPP4.png>. Last access: 23-06-20.
- Entity Framework. Entity Framework vs Dapper. 2020a. Available at <https://entityframework.net/ef-vs-dapper>. Last access: 23-06-20.
- Entity Framework. Entity Framework vs Nhibernate. 2020b. Available at <https://entityframework.net/ef-vs-nhibernate>. Last access: 23-06-20.
- Francis, J. The effects of technology on student motivation and engagement in classroom-based learning. 2017. Available at <https://dune.une.edu/cgi/viewcontent.cgi?article=1120&context=theses>. Last access: 23-06-20.
- Freire, M. Plantilla de iw curso 2019-20. 2020. Available at <https://github.com/manuel-freire/iw1920>, . Last access: 23-06-20.
- Gee, J. Learning by design: Good video games as learning machines. *E-learning*, vol. 2, 2005. Available at [https://www.researchgate.net/publication/253367747\\_Learning\\_by\\_Design\\_Good\\_Video\\_Games\\_as\\_Learning\\_Machines](https://www.researchgate.net/publication/253367747_Learning_by_Design_Good_Video_Games_as_Learning_Machines). Last access: 23-06-20.
- Groh, F. Gamification: State of the art definition and utilization. *Proceedings of the 4th Seminar on Research Trends in Media Informatics*, páginas 39-46, 2012. Available at <https://www.semanticscholar.org/paper/Gamification-%3A-State-of-the-Art-Definition-and-Groh/0d3a21a0c144a3b02abd42df59cc425b38620bf2>. Last access: 23-06-20.
- H5P. H5P. 2020. Available at <https://h5p.org/>. Last access: 23-06-20.
- Hainey, T., Connolly, T., Boyle, E., Wilson, A. y Razak, A. A systematic literature review of games-based learning empirical evidence in primary education. *Computers & Education*, vol. 102, 2016. Available at [https://www.researchgate.net/publication/307992747\\_A\\_systematic\\_literature\\_review\\_of\\_games-based\\_learning\\_empirical\\_evidence\\_in\\_primary\\_education](https://www.researchgate.net/publication/307992747_A_systematic_literature_review_of_games-based_learning_empirical_evidence_in_primary_education). Last access: 23-06-20.
- IMS Global. Learning Tools Interoperability | IMS Global Learning Consortium. 2020. Available at <http://www.imsglobal.org/activity/learning-tools-interoperability#ltiresources>. Last access: 23-06-20.



- JetBrains. Java 2019 - The state of Developer Ecosystem in 2019 Infographic. 2019. Available at <https://jetbrains.com/lp/devecosystem-2019/java/>. Last access: 23-06-20.
- Jia, Y., Liu, Y., Yu, X. y Volda, S. Designing leaderboards for gamification: Perceived differences based on user ranking, application domain, and personality traits. En , páginas 1949–1960. 2017. Available at [https://www.researchgate.net/publication/316651227\\_Designing\\_Leaderboards\\_for\\_Gamification\\_Perceived\\_Differences\\_Based\\_on\\_User\\_Ranking\\_Application\\_Domain\\_and\\_Personality\\_Traits](https://www.researchgate.net/publication/316651227_Designing_Leaderboards_for_Gamification_Perceived_Differences_Based_on_User_Ranking_Application_Domain_and_Personality_Traits). Last access: 23-06-20.
- Kayımbaşıoğlu, D., Oktekin, B. y Hacı, H. Integration of gamification technology in education. *Procedia Computer Science*, vol. 102, 2016. Available at <https://www.sciencedirect.com/science/article/pii/S1877050916326400>. Last access: 23-06-20.
- Korn, O., Funk, M. y Schmidt, A. Towards a gamification of industrial production. a comparative study in sheltered work environments. En . 2015. Available at [https://www.researchgate.net/publication/280087696\\_Towards\\_a\\_Gamification\\_of\\_Industrial\\_Production\\_A\\_Comparative\\_Study\\_in\\_Sheltered\\_Work\\_Environments](https://www.researchgate.net/publication/280087696_Towards_a_Gamification_of_Industrial_Production_A_Comparative_Study_in_Sheltered_Work_Environments). Last access: 23-06-20.
- Lee, J. y Hammer, J. Gamification in education: What, how, why bother? *Academic Exchange Quarterly*, vol. 15, páginas 1–5, 2011. Available at [https://www.researchgate.net/publication/258697764\\_Gamification\\_in\\_Education\\_What\\_How\\_Why\\_Bother](https://www.researchgate.net/publication/258697764_Gamification_in_Education_What_How_Why_Bother). Last access: 23-06-20.
- Legault, L. Intrinsic and extrinsic motivation. *Encyclopedia of Personality and Individual Differences*, 2016. Available at [https://www.researchgate.net/publication/311692691\\_Intrinsic\\_and\\_Extrinsic\\_Motivation](https://www.researchgate.net/publication/311692691_Intrinsic_and_Extrinsic_Motivation). Last access: 23-06-20.
- levelUp. Level up! Gamification for Moodle. 2019. Available at <https://levelup.plus/>. Last access: 23-06-20.
- Mekler, E., Brühlmann, F., Opwis, K. y Tuch, A. Disassembling gamification: The effects of points and meaning on user motivation and performance. En , páginas 1137–1142. 2013. Available at [https://www.researchgate.net/publication/312457140\\_Disassembling\\_gamification\\_The\\_effects\\_of\\_points\\_and\\_meaning\\_on\\_user\\_motivation\\_and\\_performance](https://www.researchgate.net/publication/312457140_Disassembling_gamification_The_effects_of_points_and_meaning_on_user_motivation_and_performance). Last access: 23-06-20.
- Microsoft. ASP.NET MVC Pattern | .NET. 2020a. Available at <https://dotnet.microsoft.com/apps/aspnet/mvc>. Last access: 23-06-20.
- Microsoft. ASP.NET Web APIs | Rest API's with .NET and C#. 2020b. Available at <https://dotnet.microsoft.com/apps/aspnet/apis>. Last access: 23-06-20.
- Moccozet, L., Tardy, C., Opprecht, W. y Michel, L. Gamification-based assessment of group work. 2013. Available at [https://www.researchgate.net/publication/255722750\\_Gamification-based\\_assessment\\_of\\_group\\_work](https://www.researchgate.net/publication/255722750_Gamification-based_assessment_of_group_work). Last access: 23-06-20.

- Moodle. About Moodle - MoodleDocs. 2018. Available at [https://docs.moodle.org/38/en/About\\_Moodle](https://docs.moodle.org/38/en/About_Moodle). Last access: 23-06-20.
- Moodle. Formato Aiken - MoodleDocs. 2020a. Available at [https://docs.moodle.org/all/es/Formato\\_Aiken](https://docs.moodle.org/all/es/Formato_Aiken). Last access: 23-06-20.
- Moodle. Moodle plugins directory. 2020b. Available at <https://moodle.org/plugins/?q=Gamification>. Last access: 23-06-20.
- Moodle. Moodle plugins directory: Game. 2020c. Available at [https://moodle.org/plugins/mod\\_game](https://moodle.org/plugins/mod_game). Last access: 23-06-20.
- Moodle. Moodle statistics. 2020d. Available at <https://stats.moodle.org/>. Last access: 23-06-20.
- Morris, W. 8 Popular PHP Frameworks For Web Development in 2020 | Hacker Noon. 2020. Available at <https://hackernoon.com/8-popular-php-frameworks-for-web-development-in-2020-od3f38ez>. Last access: 23-06-20.
- Morrison, B. y Disalvo, B. Khan academy gamifies computer science. páginas 39–44. 2014. Available at [https://www.researchgate.net/publication/261961212\\_Khan\\_Academy\\_gamifies\\_computer\\_science](https://www.researchgate.net/publication/261961212_Khan_Academy_gamifies_computer_science). Last access: 23-06-20.
- Raja, R. y Nagasubramani, P. Impact of modern technology in education. *Journal of Applied and Advanced Research*, vol. 3, página 33, 2018. Available at [https://www.researchgate.net/publication/325086709\\_Impact\\_of\\_modern\\_technology\\_in\\_education](https://www.researchgate.net/publication/325086709_Impact_of_modern_technology_in_education). Last access: 23-06-20.
- React JS. React – Una biblioteca de JavaScript para construir interfaces de usuario. 2020. Available at <https://es.reactjs.org/>. Last access: 23-06-20.
- Richter, G., Raban, D. y Rafaeli, S. *Studying Gamification: The Effect of Rewards and Incentives on Motivation*, páginas 21–46. 2015. Available at [https://www.researchgate.net/publication/321063416\\_The\\_Effect\\_of\\_Gamification\\_on\\_Motivation\\_and\\_Engagement](https://www.researchgate.net/publication/321063416_The_Effect_of_Gamification_on_Motivation_and_Engagement). Last access: 23-06-20.
- Sanchez-Mena, A. y Marti-Parreño, J. Gamification in higher education: Teachers' drivers and barriers. 2016. Available at [https://www.researchgate.net/publication/312106374\\_Gamification\\_in\\_Higher\\_Education\\_Teachers%27\\_Drivers\\_and\\_Barriers](https://www.researchgate.net/publication/312106374_Gamification_in_Higher_Education_Teachers%27_Drivers_and_Barriers). Last access: 23-06-20.
- Slant. Slant - 13 Best PHP ORMs as of 2020. 2020. Available at <https://www.slant.co/topics/5639/~php-orms>. Last access: 23-06-20.
- Solis, A. Gamified Design Review: A In-depth Analysis Of Duolingo. 2015. Available at <https://www.gamification.co/2015/08/12/gamified-design-review-a-in-depth-analysis-of-duolingo/>. Last access: 23-06-20.
- Spring. spring-overview.png. 2016. Available at <https://docs.spring.io/spring/docs/4.3.x/spring-framework-reference/html/images/spring-overview.png>. Last access: 23-06-20.

- Sun-Lin, H.-Z. y Chiou, G.-F. Effects of gamified comparison on sixth graders' algebra word problem solving and learning attitude. *Educational Technology and Society*, vol. 22, páginas 120–130, 2019. Available at [https://www.researchgate.net/publication/332540312\\_Effects\\_of\\_gamified\\_comparison\\_on\\_sixth\\_graders%27\\_algebra\\_word\\_problem\\_solving\\_and\\_learning\\_attitude](https://www.researchgate.net/publication/332540312_Effects_of_gamified_comparison_on_sixth_graders%27_algebra_word_problem_solving_and_learning_attitude). Last access: 23-06-20.
- W3Techs. PHP vs. ASP.NET vs. Java usage statistics, June 2020. 2020. Available at <https://w3techs.com/technologies/comparison/pl-aspnet,pl-java,pl-php>. Last access: 23-06-20.
- Watson-Nolan, A. The Front-End Tooling Survey 2019 - Results. 2019. Available at <https://ashleynolan.co.uk/blog/frontend-tooling-survey-2019-results#experience-general>. Last access: 23-06-20.
- Wikimedia Commons. Maslow's hierarchy. 2006. Available at [https://commons.wikimedia.org/wiki/File:Maslow%27s\\_hierarchy\\_of\\_needs.svg](https://commons.wikimedia.org/wiki/File:Maslow%27s_hierarchy_of_needs.svg). Last access: 23-06-20.
- Wikimedia Commons. Iterative development model. 2014. Available at [https://commons.wikimedia.org/wiki/File:Iterative\\_development\\_model.svg](https://commons.wikimedia.org/wiki/File:Iterative_development_model.svg). Last access: 23-06-20.
- Zichermann, G. y Cunningham, C. *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. O'Reilly Media, Inc., 1st edición, 2011. ISBN 1449397670.
- Zmaranda, D., Pop-Fele, L.-L., Györödi, C., Gyorodi, R. y Pecherle, G. Performance comparison of crud methods using net object relational mappers: A case study. *International Journal of Advanced Computer Science and Applications*, vol. 11, 2020. Available at [https://www.researchgate.net/publication/339022904\\_Performance\\_Comparison\\_of\\_CRUD\\_Methods\\_using\\_NET\\_Object\\_Relational\\_Mappers\\_A\\_Case\\_Study](https://www.researchgate.net/publication/339022904_Performance_Comparison_of_CRUD_Methods_using_NET_Object_Relational_Mappers_A_Case_Study). Last access: 23-06-20.

